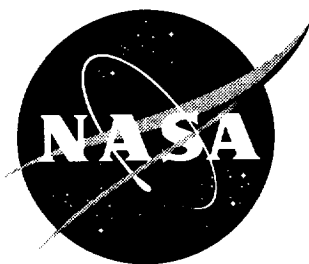


NASA/CR-2001-211244



Pathfinder Photogrammetry Research for Ultra-Lightweight and Inflatable Space Structures

*Louis Roy Miller Giersch
Joint Institute for Advancement of Flight Sciences
The George Washington University*

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Cooperative Agreement NCC1-01017

November 2001

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Abstract

The defining characteristic of ultra-lightweight and inflatable space structures is that they are both very large and very low mass. This makes standard contacting methods of measurement (e.g. attaching accelerometers) impractical because the dynamics of the structure would be changed by the mass of the contacting instrument. Optical measurements are therefore more appropriate. Photogrammetry is a leading candidate for the optical analysis of gossamer structures because it allows for the measurement of a large number of points, is amenable to time sequences, and offers the potential for a high degree of accuracy. The purpose of this thesis is to develop the methodology and determine the effectiveness of a photogrammetry system in measuring ultra-lightweight and inflatable space structures. The results of this thesis will be considered in the design of an automated photogrammetry system for the 16m-diameter vacuum chamber at the NASA Langley Research Center.

Acknowledgements

The author greatly appreciates the support of those at the George Washington University JIAFS program and the NASA Langley Research Center who have made this research possible. Dr. Paul Cooper and Richard Pappa deserve particular thanks for their continual guidance and support.

Table of Contents

Abstract	iii
Acknowledgements	iv
Table of Contents	v
List of Figures	viii
List of Tables	ix
Chapter 1: Introduction	1
1-1: Ultra-Lightweight and Inflatable Space Structures	1
1-2: A Brief History of Photogrammetry	1
1-3: Research Summary	2
Chapter 2: Photogrammetric Measurement of the 5m Concentrator	4
2-1: Overview of Photogrammetry	4
2-2: Overview of the 5m Concentrator	6
2-3: Camera Description	7
2-4: Camera Calibration	8
2-5: Measurement Planning	13
2-6: Taking the Photographs	15
2-7: Importing the Photographs into the Photogrammetry Software	16
2-8: Target Marking	17
2-9: Target Referencing	17
2-10: Processing the Data (Bundle Adjustment)	18
2-11: Exporting the Three-Dimensional Coordinate Data	19
2-12: Precision of 5m Concentrator Measurements	19
2-13: Paraboloid Fitting	22

2-14: The Effects of Blooming on Precision	23
2-15: Conclusions on the 5m concentrator measurement	25
Chapter 3: Experiments in Videogrammetry of Deploying Structures	26
3-1: Introduction to Videogrammetry	26
3-2: Overview of Poly tubing Structures	26
3-3: Experimental Setup of VMD2Cam System	27
3-4: Single-Column Tests	29
3-5: Packing of the Single-Column Test Articles	29
3-6: The Cuff-Fold	30
3-7: Tripod Tests	32
3-8: Retro-Reflective vs. Flat White Targets	34
3-9: Dot Projection and Membrane Materials	36
3-10: Matlab/PhotoModeler DDE System	40
Chapter 4: Recommendations for Future Work	42
4-1: Increased Number of Cameras and Software Development	42
4-2: Coded Targets	44
4-3: Inflation Control	45
Chapter 5: Concluding Remarks	46
References	49
Appendix	A-1
Digitia script used for DC290 settings	A-1
Matlab code “opendata2.m” used to read xyz and precision data exported from PhotoModeler	A-1

Matlab function “fitparabola.m” used to fit xyz data to a parabolic surface	A-3
Matlab function “ddedemo.m” used to demonstrate viability of	
Matlab/PhotoModeler DDE system	A-6
Matlab function “RealSpace” used to simulate the camera views of a	
dynamic, 3d object	A-8
Matlab function “k_cam1” used to simulate the K1 distortion of a	
DC290 camera	A-9
Matlab function “LGVS3,” A prototype target tracking videogrammetry	
system using Matlab/PhotoModeler DDE	A-10

List of Figures

Figure 1: The 5m concentrator mounted in the 16m vacuum chamber	7
Figure 2: The projection and distortion of a three-dimensional scene.....	8
Figure 3: Distortion of points as described by lens distortion parameters	9
Figure 4: Photos used by PhotoModeler in a “field calibration”	11
Figure 5: K1 distortion typical of DC290 cameras	13
Figure 6: Triangulating the two-dimensional location of a point	14
Figure 7: Camera locations used for photographing the 5m concentrator	15
Figure 8: Example of an underexposed photograph of the 5m concentrator.....	16
Figure 9: The sub-pixel target marker.....	17
Figure 10: Examples of Z-fold and roll.....	29
Figure 11: Example of Cuff-fold	30
Figure 12: Downward deployment tests data from a Cuff-folded column.....	32
Figure 13: Configurations used in attempts to measure all three tripod legs	33
Figure 14: Tripod transitioning from a packed state to two deployed states.....	34
Figure 15: Experimental setup of the target comparison tests.....	35
Figure 16: Variation in amplitude as a function of angle and lighting conditions	35
Figure 17: Dot projection (above) and resulting wire frame data (below) of a chair	36
Figure 18: A microscopic perspective of how light reflects.....	38
Figure 19: Experimental setup of the dot projection cases.....	39
Figure 20: Layout of a Matlab/PhotoModeler-based videogrammetry system.....	41

List of Tables

Table 1: DC290 settings	11
Table 2: Camera Parameters for DC290 #1.....	12
Table 3: Camera parameters used for photogrammetry measurements	12
Table 4: Precisions of the 5m concentrator measurements in inches.....	20
Table 5: Results of the paraboloid fitting algorithms.....	23
Table 6: Precision of each case and the corresponding flash intensity reductions	24
Table 7: Qualitative description of material properties.....	38
Table 8: Quantitative contrast measurements.....	39

Chapter 1 : Introduction

In this chapter, a description of ultra-lightweight and inflatable space structures is presented, as is a brief history of photogrammetry. A summary of the research discussed in this thesis is also provided.

1-1: Ultra-Lightweight and Inflatable Space Structures

Ultra-lightweight and inflatable structures hold immense potential for space-based applications. These structures have very low densities, thus reducing the payload mass requirements for launch vehicles. They can deploy from an initially small, packed volume, thus reducing payload volume requirements. These gossamer structures can be deployed to great scales, allowing exceptionally large volumes, areas, and lengths to be employed in space structures. Within the NASA Gossamer Spacecraft Initiative, concepts for inflatable habitats, solar and optical concentrators, antennas, solar sails, and solar shades are under study (1,2,3).

1-2: A Brief History of Photogrammetry

Ironically, the mathematical theory behind photogrammetry has existed longer than photography. In 1715, Dr. Brook Taylor published the book *Linear Perspective* dealing with the mathematical projection of a three-dimensional scene onto a two-dimensional plane. In 1759, J.H. Lambert suggested that the principles of perspective could be used to produce accurate maps (4). This would in fact become the primary application of photogrammetry, but its use would have to wait until practical photography had been developed.

In 1839, Louis Daguerre publicized his technique for direct photography using metal plates coated with light-sensitive silver iodide. In 1849, Colonel Aime Laussedat of

the French Army Corps of Engineers directed the first experiments in using photogrammetry for topographic mapping (4). Colonel Laussedat experimented with both terrestrial photographs and aerial photographs taken from balloons and kites, but the practical difficulties then associated with aerial photography limited this branch of his work. The invention of the airplane by the Wright brothers in 1902 provided the means for aerial photogrammetry to develop. Aerial photographs were used primarily for reconnaissance in World War I, but it was during World War II that aerial photogrammetry was used on a massive scale to meet the urgent demand for maps.

While still used as a tool in the production of maps, photogrammetry is finding applications in such diverse fields as tool inspection, crime scene investigation, and motion analysis (4). The use of photogrammetry in map making is known as *topographic photogrammetry*, while the use of photogrammetry in other fields such as those mentioned above is known as *non-topographic* or *close-range photogrammetry*. While the applications of photogrammetry are diverse, the underlying techniques are common.

1-3: Research Summary

The purpose of this thesis is to develop the methodology and ascertain the effectiveness of using photogrammetry to measure ultra-lightweight and inflatable space structures. Experiments relevant to the measurement of the static shape and the deployment dynamics of various structures similar to ultra-lightweight and inflatable space structures were conducted.

The process of making photogrammetric measurements of a 5-meter diameter inflatable solar concentrator is described. This concentrator was photogrammetrically measured to determine what precisions could be obtained and how these precisions varied

with camera resolution and the number of images used in the measurement. Experiments in measuring the deployment of structures composed of inflatable columns are explained, with an emphasis give to the techniques used instead of measured results. Discussions of general experimental methods applicable to both static and dynamic measurements are given throughout. Based on the research experience, recommendations for future work are made and concluding remarks on the potential use of photogrammetry in measuring ultra-lightweight and inflatable space structures are given.

Chapter 2 : Photogrammetric Measurement of the 5m Concentrator

In this chapter, a brief overview of photogrammetry is given, followed by a description of the ultra-lightweight 5m diameter solar concentrator. The process of measuring the 5m concentrator using photogrammetry is then described, and the results are presented.

2-1: Overview of Photogrammetry

Photogrammetry is the science of analyzing photographs to obtain accurate measurements of physical objects. A photograph is the projection of a three-dimensional scene onto a two-dimensional plane, such as a photographic film or a charge-coupled device (CCD)*. The foundation of photogrammetry is triangulation, in which two or more photographs are used to reconstruct the three-dimensional coordinates of the photographed scene. Triangulation requires knowledge of the orientation of the photographic planes with respect to the scene, and so the positions and orientations of the cameras must be determined (5, 6). This information, as well as the three-dimensional coordinates of the scene can be calculated iteratively and simultaneously using what is known as a bundle adjustment algorithm (7).

The projection of the scene onto the photographic plane will be affected by not only the location and orientation of the camera, but also by the physical properties of the camera itself. These properties, such as focal length and lens distortion, are determined by “calibrating” the camera. Camera calibration can be done by creating a photogrammetric model of a scene with known coordinates, such as a grid projected onto a flat wall. This is known as a “field calibration.”

* Charge-coupled devices (CCDs) are used in video and digital cameras to capture and record light.

When photogrammetric measurements are to be made of a scene, it is important to choose camera locations and orientations that will yield the most accurate results. Calculation of three-dimensional coordinates requires images of a scene from two or more cameras taken at convergent camera angles. Angular separation, or the angle between two cameras and the center of the scene, is of considerable importance. An angular separation of 90 degrees is optimal to minimize the angular error sensitivity of the cameras, but an angle half this size is acceptable: angular separations less than 15 degrees or greater than 165 degrees should be avoided. The locations and orientations of the cameras used to photograph the scene are automatically computed when the photogrammetric measurements are calculated, and so it is not necessary to measure and record these location while taking the photographs. The scale of a scene cannot be determined from photographs that do not contain objects of known size, and so “scale bars” (bars of known length) are included in all the scenes imaged in this thesis.

Once the photographs have been taken, the images are loaded into a photogrammetry software package. The software package used in this thesis is PhotoModeler Pro from Eos Systems, Inc., which is a consumer-grade photogrammetry package. Within the software, the camera calibration parameters are entered and points of interest are marked on the photographs. Corresponding points on different images are then “referenced” to each other. Referencing tells the software that point A in picture 1 is the same physical point as point B in picture 2. When a minimum number of points have been marked and referenced (approximately 10 points per photo), the bundle adjustment can be executed and the three-dimensional coordinates of the referenced points as well as the camera locations and orientations are calculated. Additional points can be marked and

referenced and the bundle adjustment re-executed until all points of interest have been measured.

When the three-dimensional coordinates of all points of interest in the scene have been calculated, this data can be exported and studied with other software packages. In this thesis, the data was exported and studied in detail using MATLAB from The MathWorks, Inc.

2-2: Overview of the 5m Concentrator

The test article under study in this chapter is an inflatable parabolic solar concentrator manufactured by SRS Technologies in Huntsville, Alabama (Figure 1). The concentrator consists of two inflatable structures: the parabolic lenticular and the torus. The lenticular has a transparent convex dome covering a highly reflective concave parabolic membrane 5m in diameter. The outer diameter of the torus is 6.5m, with a cross-sectional diameter of 0.6m. The torus supports the lenticular with a series of thin cords attaching the perimeter of the lenticular to the torus. The total mass of the structure is roughly 4 kg. Similar structures are being investigated for use in space-based solar power generation, solar thermal propulsion, radio and optical astronomy, and antennas (1, 8, 9).

Small circular retro-reflective targets have been placed on the lenticular for use in photogrammetry studies. Larger square retro-reflective targets have been placed on both the lenticular and the torus for use in separate laser vibrometry studies. No photogrammetry targets were placed on the torus since the focus of this study is determination of the shape of the lenticular. As a matter of convention, this parabolic concentrator (both lenticular and torus) will be referred to as the 5m concentrator, and

when mention of photogrammetric measurements are made to the 5m concentrator, it is understood that this refers to only the lenticular.

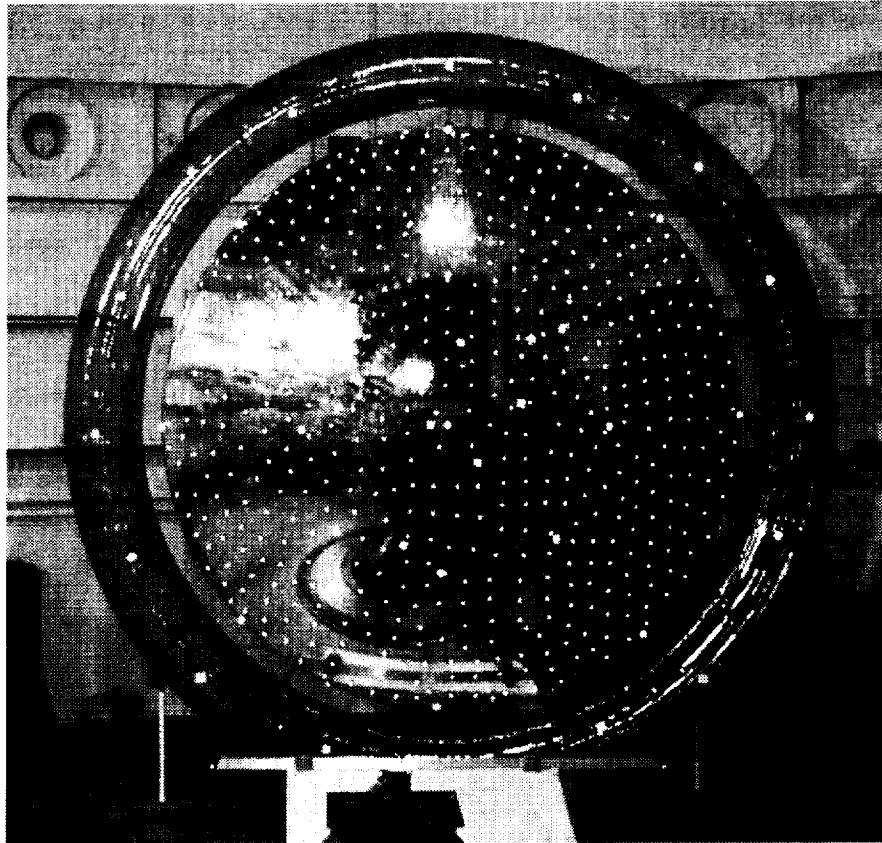


Figure 1: The 5m concentrator mounted in the 16m vacuum chamber. The rear, convex surface of the concentrator is shown, and this is the surface measured in this research.

There are eight steps involved in making photogrammetric measurements. These steps are here described in the context of measuring the 5m concentrator (10).

2-3: Camera Description

The two primary types of digital still cameras used in this research are the Kodak DC290 and the Kodak DC4800. The DC290 has a resolution of 1792 x 1200 pixels (≈ 2 megapixel) with pixels approximately 4.2 microns square. The DC4800 has a resolution of 2160 x 1440 (≈ 3 megapixel) with pixels approximately 3.5 microns square. For both cameras, all the photograph used to make photogrammetric measurements and calibrations were taken with the built-in zoom lenses set to full wide.

2-4: Camera Calibration

The physical properties of a camera, such as the focal length and lens distortion, are known as the “camera parameters,” and the determination of these parameters is known as “camera calibration.” Accurate photogrammetry measurements require accurate camera calibrations. The camera parameters describe the geometry and distortion of the projection of a three-dimensional scene onto the two-dimensional CCD of the camera (Figure 2). The photogrammetry software compensates for these geometric effects if the camera parameters are known.

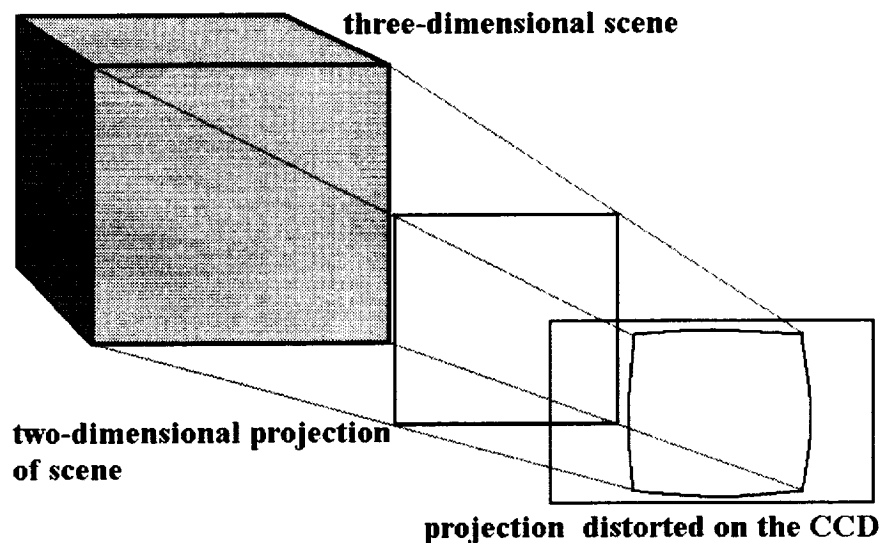


Figure 2: The projection and distortion of a three-dimensional scene onto the CCD of the camera

In an idealized pinhole camera, all of the light rays are focused through a single point in space as they pass through the camera. After the rays pass through this point, the image is flipped upside-down and left-to-right. The single point in space through which all light rays pass is known as the perspective center of the camera. The principal point of the photograph is the projection of the perspective center of the camera onto the photographic plane, and is usually near the center of the photograph. In the cameras discussed here, the camera lenses focus the light through a finite area, not a single point,

and the principal point of these cameras is at the center of the projection of this area onto the photographic plane. The exact location of the principal point must be determined by camera calibration.

The K_1 , K_2 , P_1 , and P_2 distortion parameters are quantitative measures of four lens distortion effects (Figure 3). The K_1 parameter measures the radial distortion of the lens, which creates a “barrel” or “pincushion” effect. The K_2 parameter is similar to the K_1 effect, but is a higher-order term and thus only identifiable near the edges of the photograph and is often negligible. The P_1 and P_2 parameters measure the misalignment of the lens elements.

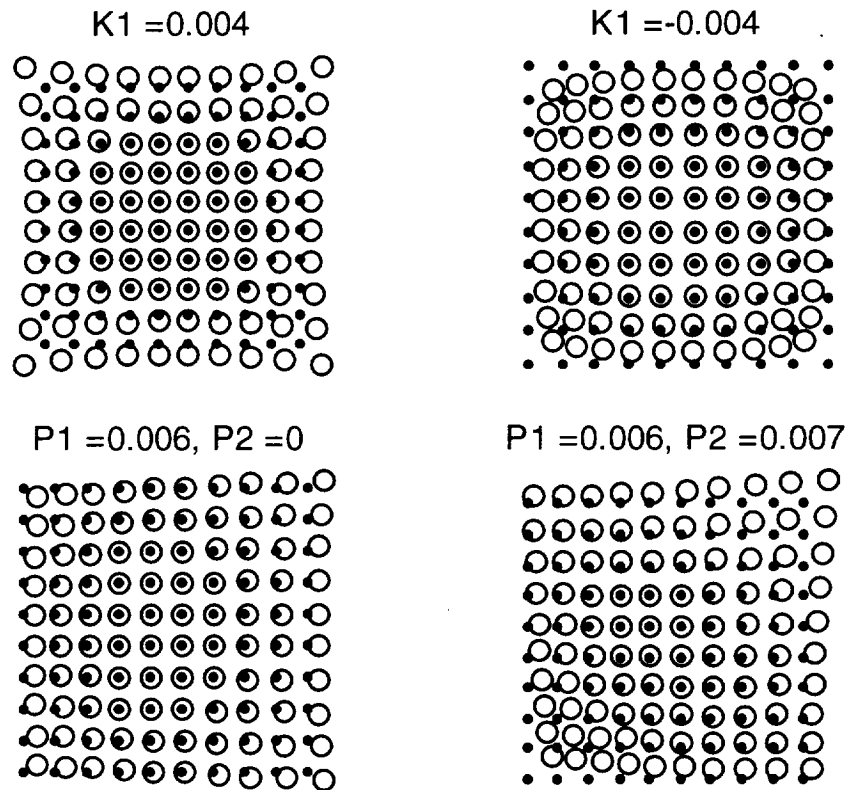


Figure 3: Distortion of points as described by lens distortion parameters. Dots represent points prior to distortion; circles represent distorted points.

The shift in the location of a projected point onto the image plane due to lens distortion is given by the following equations:

$$\Delta x = K_1 x r^2 + K_2 x r^4 + P_1(r^2 + 2x^2) + 2P_2 xy,$$

$$\Delta y = K_1 y r^2 + K_2 y r^4 + P_2(r^2 + 2y^2) + 2P_1 xy,$$

$$\text{with } r^2 = x^2 + y^2,$$

where Δx and Δy are the change in horizontal and vertical position of the point located at x and y , with the origin of the coordinate system located at the principal point of the photograph.

PhotoModeler estimates camera parameters using photographs of a grid pattern projected onto a flat wall (Figure 4). This is known as a “field calibration.” The photographs used in the field calibration are taken from various locations and orientations. There is no need to measure the locations and orientations of the camera when taking these photographs because this information is automatically calculated by the calibration software. Once these images have been imported into PhotoModeler, the user uses the mouse to mark four reference points located in the corners of each photo. Once the reference points have been marked, the hand-measured distance between two of the reference points on a diagonal of the projected pattern is input into the program. The program then begins an automated process to determine the camera parameters. The focal length, CCD size (also known as the format size), principal point location, and lens distortion parameters are thus determined.

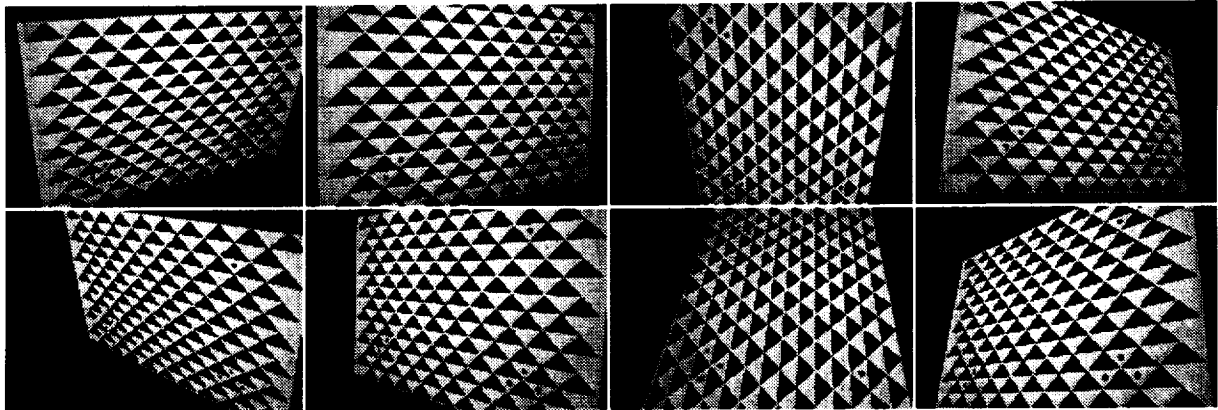


Figure 4: Photos used by PhotoModeler in a “field calibration”

Camera settings such as the manual focus distance and the zoom setting affect camera parameters. It is therefore necessary to insure that the camera settings are the same each time photographs are taken. The camera settings used to make measurements must also be the same as the settings used to make the camera calibration photographs. To this purpose, a script for the “Digita” programming language was written for the DC290 cameras. This script (see appendix) automatically changed the camera settings when the camera is turned on from the default values used to the desired settings used for the photogrammetry measurements. The settings used for the DC290 are given in Table 3. A limited number of photographs were also taken with the DC4800 digital camera. The DC4800 does not use Digita scripts, and so the settings had to be changed manually every time the camera was used.

Table 1: DC290 settings

Resolution:	1792 x 1200 pixels
Quality:	Best (least JPEG compression)
Zoom:	Full wide
White balance:	Automatic
Exposure Compensation:	0
Manual focus distance:	5m

The accuracy of the camera parameters is critical for obtaining accurate photogrammetry measurements. Therefore, each of the four Kodak DC290 digital cameras used in this research was calibrated multiple times over the course of several days. The variation in camera parameters for one of the DC290 cameras is shown in Table 1. The average camera parameters for each DC290 were used for the photogrammetry measurements, and these are shown in Table 2. A computer simulation of the distortion described by the K1 parameter typical of the DC290 used is shown in Figure 5. The K1 parameter is the dominant parameter, as can be seen in Table 1.

Table 2: Camera Parameters for DC290 #1

Camera 1	Focal Length (mm)	Format Size W (mm)	Format Size H (mm)	Principal Point X (mm)	Principal Point Y (mm)	K1	K2	P1	P2
Day 1	8.2282	7.5593	5.04	3.8134	2.4521	1.686E-3	-4.817E-5	-3.897E-5	-3.13E-4
Day 2a	8.2086	7.5324	5.04	3.8383	2.5508	1.824E-3	-4.646E-5	7.049E-5	-7.469E-5
Day 2b	8.1990	7.5328	5.04	3.8549	2.5442	1.733 E-3	-3.98E-5	-2.056E-5	-5.871E-5
Day 2c	8.1872	7.5324	5.04	3.8226	2.5650	1.698 E-3	-3.999E-5	2.966E-5	1.604E-6
Day 3a	8.2019	7.5333	5.04	3.8514	2.5460	1.788 E-3	-4.725E-5	-4.744E-5	-7.821E-5
Day 3b	8.2008	7.5330	5.04	3.8470	2.5437	1.717 E-3	-3.892E-5	-1.291E-5	-1.059E-4
Day 4a	8.1975	7.5360	5.04	3.8494	2.5353	1.753 E-3	-4.249E-5	2.941E-5	-6.11E-5
Day 4b	8.2023	7.5342	5.04	3.8517	2.5433	1.744 E-3	-4.096E-5	2.141E-7	-9.69E-5
Day 4c	8.2028	7.5367	5.04	3.8484	2.5490	1.721 E-3	-4.017E-5	2.684E-5	-5.706E-5
Mean:	8.2031	7.5367	5.04	3.8419	2.5366	1.740 E-3	-4.269E-5	4.082E-6	-9.377E-5
Std. Deviation:	0.0110	0.0086	0	0.0145	0.0327	4.353E-5	3.610E-6	3.815E-5	8.772E-5

Table 3: Camera parameters used for photogrammetry measurements

	Focal Length (mm)	Format Size W (mm)	Format Size H (mm)	Principal Point X (mm)	Principal Point Y (mm)	K1	K2	P1	P2
DC290 #1 Mean:	8.203	7.537	5.04	3.842	2.537	0.00174	-4.27E-5	0.41E-5	-9.38E-5
DC290 #2 Mean:	8.206	7.532	5.04	3.888	2.349	0.00155	-4.10E-5	-13.7E-5	-4.19E-5
DC290 #3 Mean:	8.181	7.531	5.04	3.822	2.420	0.00163	-4.47E-5	-0.97E-5	0.03E-5
DC290 #4 Mean:	8.232	7.533	5.04	3.843	2.504	0.00162	-4.18E-5	-2.46E-5	-9.18E-5

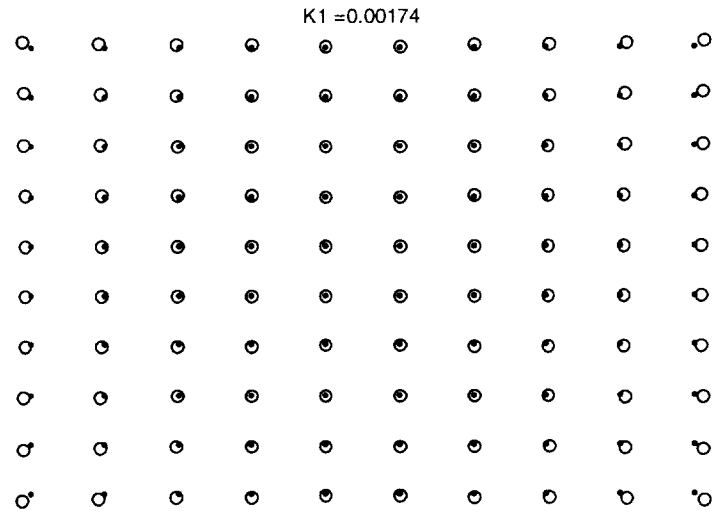


Figure 5: K1 distortion typical of DC290 cameras

2-5: Measurement Planning

The number of photographs taken and the horizontal and vertical angular separations of the camera locations will affect the accuracy of a photogrammetric measurement. It is also necessary to include at least one scale bar in the photographs. It is therefore essential to plan the measurements before any photographs are taken. The 5m concentrator was mounted horizontally (the line of symmetry of the 5m concentrator was horizontal) for all photogrammetry experiments discussed in this thesis. Two scale bars, one vertical bar located to the left of the 5m concentrator, and one horizontal bar located below the 5m concentrator, were added to the photographed scene for scaling purposes.

The importance of angular separation can be seen in the following example. Suppose we are trying to find the two-dimensional location of a point by triangulation using a linear CCD array (Figure 6). The linear CCD array can resolve the projection of the point onto the array to a certain precision (e.g. one pixel). In order to triangulate the two-dimensional location of the point, the linear CCD array must image the point from

two locations with a non-zero angular separation. If we use an angular separation of 90° , we achieve a much more precise measurement of the XY location of the point than if we use a separation of 10° . That is, with a 90° angular separation, any uncertainty in the knowledge of the camera pointing direction translates into much less triangulation uncertainty, particularly in the y direction.

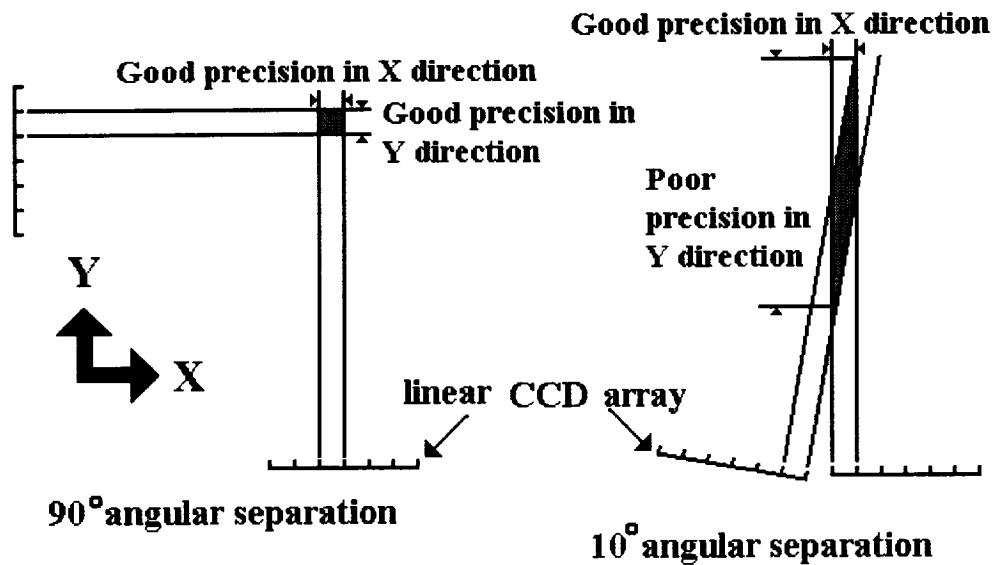


Figure 6: Triangulating the two-dimensional location of a point using a linear CCD array

Because it is desirable to have the scene appear as large as possible in the photographs (for the same reason it is desirable to have high-resolution photographs), the cameras were placed as close as possible to the 5m concentrator. Using the DC290 cameras with the zoom lens set to the full wide position, the cameras had to be at least 8m from the 5m concentrator for the entire 5m concentrator and the scale bars to be visible in each photo. Up to 9 photographs were taken per photogrammetry measurement. To achieve the desired vertical angular separation, photos were taken from floor level, at the top of a stepladder, and on a 6m-high scaffold. The floor and the height of the available scaffold prohibited larger vertical separations. Horizontal separation was achieved by moving the floor, ladder, and scaffold locations approximately 4m to the left

and right of the center (Figure 7). Larger horizontal separations were not used because the far edge of the convex concentrator surface would not be visible in the resulting pictures. This arrangement yielded a maximum vertical angular separation of 33.6 degrees and a maximum horizontal separation of 58.5 degrees. The angular separation between diagonally opposed cameras (e.g. the top left and bottom right cameras) is 67.5 degrees. In the case of 4 photograph measurements, only the top left, top right, bottom left, and bottom right camera locations were used.

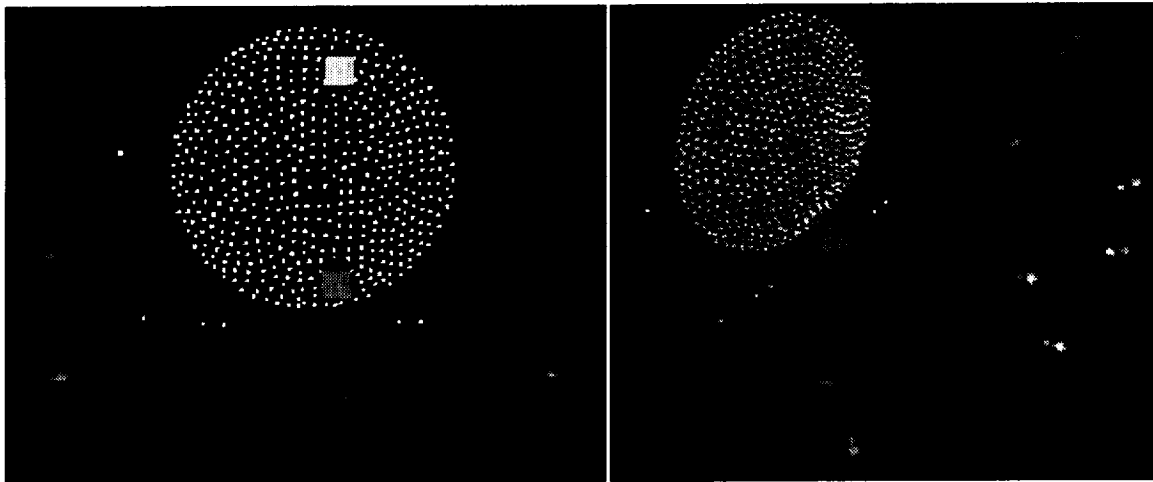


Figure 7: Camera locations used for photographing the 5m concentrator. At left, a front view is shown. At right, a skewed view is shown.

2-6: Taking the Photographs

To aid in the accurate marking of points, high contrast photographs are desirable. The contrast between the retro-reflective targets and the 5m concentrator was maximized by using the built-in camera flash with the ambient lights turned off. The resulting photographs are underexposed with the retro-reflective targets appearing as bright white ellipses on a black background (Figure 8). This type of photograph is ideal for photogrammetric measurements because the points of interest (in this case, the retro-reflective targets) are clearly distinguished from the rest of the scene.

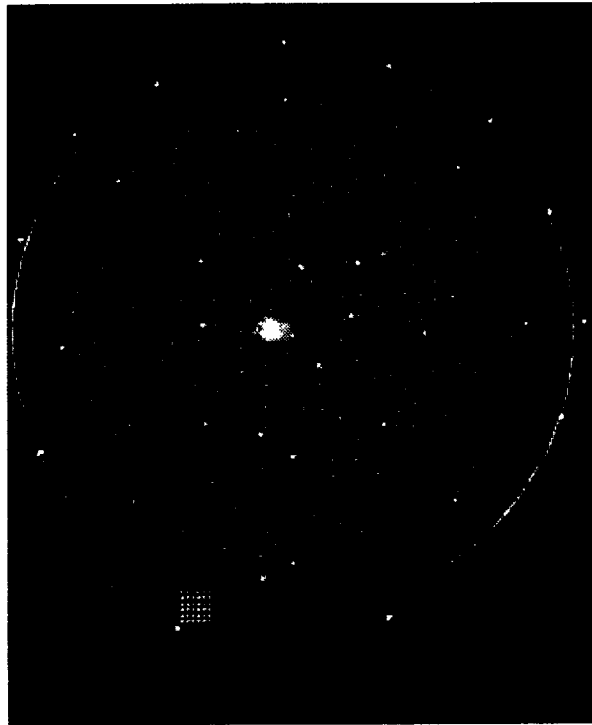


Figure 8: Example of an underexposed photograph of the 5m concentrator used in photogrammetry

An important feature of the bundle adjustment algorithm used in photogrammetry is that it will automatically determine the locations and orientations of the cameras used to make the photogrammetry measurements. This removes the need for the photographer to measure the camera locations while taking photographs. By using a short exposure time, the camera can be hand-held by the photographer and not affected by slight vibration, eliminating the need to use vibrations isolation equipment or tripods for the cameras.

2-7: Importing the Photographs into the Photogrammetry Software

The Kodak digital cameras store image files on CompactFlash memory cards. These cards are solid-state devices with storage space ranging from 4 to over 200 megabytes. Each photograph is stored as a JPEG file approximately 500-kilobytes in size. Peripheral card readers can be installed on computers so that the transfer of image files

from camera to computer is as simple as removing the CompactFlash card from one and inserting it into the other. The files are typically transferred from the card to the hard disk of the computer and from there are imported into the photogrammetry software, PhotoModeler. When the images are first loaded in PhotoModeler, the user associates each picture with a camera calibration file corresponding to the particular camera used to photograph the image.

2-8: Target Marking

Once the images are imported into PhotoModeler, the retro-reflective targets are marked. Each circular target appears as an ellipse whose aspect ratio varies with the relative orientation of the camera. Each ellipse is approximately 5 to 10 pixels in size in the photographs. Using the sub-pixel marking function of the PhotoModeler software, the location of the center of each target is determined with an accuracy of 1/10 of a pixel or better. The user defines a rectangular perimeter around the target and PhotoModeler determines the center of the target and marks and records the location. Large numbers of targets in a user-selected area of the image can also be marked with sub-pixel accuracy using an automatic target marking function in PhotoModeler. The sub-pixel target marker works similarly to a curve-fitting algorithm (Figure 9).



Figure 9: The sub-pixel target marker

2-9: Target Referencing

Target referencing means identifying targets on multiple images that represent the same physical target on the structure. A small number of targets (typically between ten and twenty) must be referenced manually before the locations and orientations of the photographs can be determined by processing the data using the bundle adjustment method. Thus, not only are the three-dimensional locations of the initially small set of referenced targets calculated, but also the three-dimensional location and orientation of the cameras used to photograph the scene are found. This allows the PhotoModeler software to then automatically reference the remaining points. This feature, along with automatic target marking, allow for more than 500 targets on the 5m concentrator to be marked and referenced on multiple photographs much more efficiently than if these tasks were to be done manually, as was the case with the original version of the PhotoModeler software used in this research.

2-10: Processing the Data (Bundle Adjustment)

Processing the data with the bundle adjustment method in PhotoModeler is a largely transparent procedure for the user. When enough targets have been marked, the data can be processed and the bundle adjustment executed. The bundle adjustment does several things iteratively: it calculates the three-dimensional locations of the referenced targets, it calculates the location and orientation of the photographs, and it can adjust the camera parameters to obtain results that are more consistent. This is all accomplished in a nonlinear least-squares solution with $2 \times N \times n$ equations in $6 \times N + 3 \times n + 8 \times c$ unknowns, where N is the number of cameras, n is the number of points, and c is the number of cameras being calibrated. For example, if one camera is used in 4 different locations to photograph 500 points, there are 4000 equations and 1532 unknowns. There are several

variations of the bundle adjustment, with various options and levels of sophistication (8). Because the position and orientation of the photographs affects the calculated location of the points, and because the location of the points is used to determine the position and location of the photographs, the algorithm runs through successive iterations until a specified precision or maximum number of iterations is reached.

2-11: Exporting the Three-Dimensional Coordinate Data

Exporting the data from PhotoModeler is straightforward. The calculated three-dimensional locations of each referenced target, along with other data such as the precision of the calculated location measurement, can be exported in a number of formats including text files. For the data analyses discussed here, the PhotoModeler data were exported to a text file and subsequently read into Matlab.

2-12: Precision of 5m Concentrator Measurements

The precisions of photogrammetrically determined three-dimensional measurements are automatically calculated for every point in PhotoModeler. These precision values represent two standard deviations, giving a 95% confidence interval for that point (i.e. a 95% probability that the true point falls within the interval defined by the precision numbers assuming that bias errors are zero) (11). Each point has separate measurement precisions in the x, y, and z directions.

The 2-megapixel Kodak DC290 and the 3-megapixel Kodak DC4800 were used to make separate photogrammetric measurements of the 5m concentrator. A specialized VSTARS digital camera, with a resolution of 3070 x 2056 pixels (≈ 6 megapixels) and pixels approximately 9 microns square was used to make additional photogrammetric measurements. These various cameras were used to examine the effects of camera

resolution on measurement precision. Two sets of measurements were made with the DC290 cameras: one using four photographs and the other using nine photographs. This was done to examine the relationship between the number of photographs and the precision of the resulting measurements. The precisions of these measurements are given in Table 4. For each set of measurements, the precisions in each direction were examined by finding the maximum, minimum, mean, and standard deviation of the point precisions. In all measurements, the x direction is horizontal, the y direction is vertical, and the z direction is along the line of symmetry of the 5m concentrator. The root-sum-square of the precisions was then calculated to provide an over-all precision of each measurement method.

Table 4: Precisions of the 5m concentrator measurements in inches

Four Photo DC 290 (2 megapixel)	Min	Max	Mean	Std Dev.
X precision:	0.003	0.023	0.009	0.004
Y precision:	0.015	0.041	0.018	0.004
Z precision:	0.037	0.112	0.052	0.024
Root-sum-square:	0.040	0.121	0.056	0.025
Four Photo DC 4800 (3 megapixel)				
X precision:	0.012	0.031	0.016	0.003
Y precision:	0.023	0.047	0.029	0.004
Z precision:	0.014	0.042	0.019	0.004
Root-sum-square:	0.029	0.071	0.038	0.006
Nine Photo DC 290 (2 megapixel)				
X precision:	0.013	0.032	0.018	0.004
Y precision:	0.016	0.040	0.022	0.004
Z precision:	0.023	0.062	0.026	0.004
Root-sum-square:	0.031	0.081	0.039	0.006
Nine Photo VSTARS (6 megapixel)				
X precision:	0.003	0.024	0.006	0.001
Y precision:	0.010	0.021	0.012	0.002
Z precision:	0.021	0.069	0.031	0.007
Root-sum-square:	0.023	0.076	0.034	0.007

If the root-sum-square value of the mean precision (in bold print) is used to measure the overall precision of each set of measurements, it is seen that precision

increases with camera resolution (e.g. the 2-megapixel DC290 is less precise than the 3-megapixel DC4800 when an equal number of photographs are used). Measurements made using four DC290 photographs are less accurate than measurements made using nine DC290 photographs, which indicates that increasing the number of photographs used in making photogrammetric measurements also increases precision of the measurements. Indeed, using nine 2-megapixel DC290 photos yields precisions similar to those obtained by using only four 3-megapixel DC4800 photos. This is an important result because it allows for trades to be made between the cost per camera and the total number of cameras needed in a photogrammetry system.

The current cost of the DC290 is approximately \$700, and the cost of the DC4800 is approximately \$800. A system of nine DC290 cameras would therefore cost about \$6300, and a system of four DC4800 cameras would cost about \$3200. Although each system will have about the same precision, the four-camera DC4800 system is just over half the price of a nine-camera DC290 system, obviously making the DC4800 system preferable.

Another interesting result is that while the nine-photo VSTARS measurements (processed by PhotoModeler) are more precise than the nine-photo DC290 measurements, the increase in precision is not as great as one might expect when considering the increase in accuracy between the four-photo DC290 and four-photo DC4800 measurements. The increase in precision in the later case is approximately 50% (0.038" vs. 0.056"), corresponding to an increase in camera resolution of 50% (2-megapixel vs. 3-megapixel). However, the increase in accuracy between the nine-photo DC290 measurements and the nine-photo VSTARS measurements is only 15% (0.034"

vs. 0.039”), corresponding to an increase in camera resolution of 200% (2 megapixel vs. 6 megapixel). This may be due to limitations in the \$700 consumer-grade PhotoModeler software. Alternatively, it may be due to the apparent sizes of the targets in the photographs used to make the measurements. Both the DC290 and DC4800 image the targets such that the targets are between 6 and 10 pixels across. The VSTARS camera images the targets such that the targets are between 4 and 5 pixels across, and this small target size may lead to poor sub-pixel marking accuracy. While the reasons for the lack of significant precision improvement using the VSTARS camera are not known, it is worth mentioning that the VSTARS camera is specifically designed for use with specialized software. When the VSTARS camera is used with the VSTARS photogrammetry software, a professional system costing roughly \$150,000, the precision specification is 0.005” for a structure the size of the 5m concentrator. This is roughly ten-times as precise as the results obtainable using the commercial Kodak cameras and PhotoModeler software, but precisions of this level may not be justified given the high increase in cost.

2-13: Paraboloid Fitting

The 5m concentrator surface is designed to be parabolic. This shape allows it to act as an effective solar concentrator or antenna. It is therefore of interest to determine how well the photogrammetrically measured points describe a parabolic surface.

An analysis algorithm, “Fitparabola.m” (shown in the Appendix), was developed in Matlab to fit the xyz locations of the points measured in PhotoModeler to a parabolic surface. Once the algorithm has read the xyz data, the data must be oriented such that the z-axis of the data is aligned with the axis of symmetry of the parabolic concentrator. Once this is done, the data are fit to a surface of the form:

$$z = ax^2 + by^2 + cx + dy + e$$

The focal length of the surface is then determined as well as the root-mean-square error of the fit. An independent algorithm developed at SRS Technologies in Huntsville, Alabama (where the 5m concentrator was designed and manufactured), was used to check the results of this algorithm. The results of both algorithms for various data sets are shown in Table 5.

Table 5: Results of the paraboloid fitting algorithms

Curve Fitting Algorithm	Focal Length		RMS	
	Fitparabola.m	SRS	Fitparabola.m	SRS
Four Photo DC290	120.9013"	120.9006"	0.0612"	0.0614"
Nine Photo DC290	119.7064"	119.7026"	0.0596"	0.0598"
Four Photo DC4800	120.7605"	120.7575"	0.0559"	0.0594"
Nine Photo VSTARS	119.7441"	119.7394"	0.0551"	0.0596"

Excellent consistency can be seen between the results of both Fitparabola.m and the SRS algorithm. The 5m concentrator was designed to have a focal length of 120.0" and an RMS surface error of less than 0.0397" ($\approx 1\text{mm}$), and the photogrammetric analysis done by SRS of the 5m concentrator after construction at SRS Technologies in Huntsville, Alabama, measured an RMS surface error of 0.0096". The difference between these numbers and the ones found here may be potentially caused by either the precision of the measurements (which are on the order of the calculated RMS error) or the fact that the 5m concentrator has been deflated, packed, shipped, and re-inflated many times since the original photogrammetric analysis.

2-14: The Effects of Blooming on Precision

When photographs are taken with an intense flash, the retro-reflective targets in the scene can appear larger in the photo than they actually are. This effect is known as "blooming." The effects of blooming on the precision of photogrammetric measurements

made using sub-pixel targeting are not well known. Blooming may improve precision by effectively increasing the size of the targets and thus the precision of the sub-pixel targeting. However, blooming will not necessarily be a symmetric effect, and as such, it might add random error to the measured location, which is undesirable.

To test the effects of blooming on precision, the 5m concentrator was photographed from three camera stations using the DC290 camera. At each station four photos were taken, each at a different flash intensity. The flash was covered with an increasing number of layers of masking tape to reduce the flash intensity. Four photogrammetric measurements were then made, each corresponding to a certain flash intensity. The root-sum-square of the precisions for each case and the corresponding flash intensity reductions are given in Table 6.

Table 6: Precision of each case and the corresponding flash intensity reductions

Case:	A	B	C	D
Layers of masking tape covering the flash:	0	1	2	3
Root-sum-square max precision	0.3655"	0.5844"	0.4927"	0.4915"
Root-sum-square min precision	0.0504"	0.0553"	0.0559"	0.0560"
Root-sum-square mean precision	0.0832"	0.1052"	0.1132"	0.1125"
Root-sum-square Std Dev. precision	0.0269"	0.0427"	0.0456"	0.0456"

It is seen that Case A is most precise and has the most intense flash, implying that in this case blooming may have improved the precision of the measurements. Not only is the mean value of precision for Case A the lowest, but so is the maximum value and standard deviation. It is interesting to note that Case C and D are nearly identical in precision. This may be due to the fact that at these flash intensities, blooming has been eliminated. Repeating this experiment with a more-accurate way of controlling the flash intensity (e.g. an external, adjustable flash unit) may provide a clearer picture of what effects blooming has on precision. The geometry of a particular test article may also

affect any asymmetric blooming, and so an adjustable flash unit is recommended for the 16m vacuum chamber system.

2-15: Conclusions on the 5m concentrator measurement

It has been shown that the 5m concentrator can be effectively measured using commercial digital cameras and photogrammetry software. These measurements are precise enough to allow comparisons to be made between the measured shape of the structure and the engineering specifications such as the focal length.

Chapter 3 : Experiments in Videogrammetry of Deploying Structures

In this chapter, experiments in measuring the deployment of structures composed of inflatable columns are discussed. A comparison of different types of targets is also given, as is an evaluation of candidate membrane materials to be measured using a technique known as “dot projection” in which physical adhesive targets are replaced with targets optically projected onto the test article. An experiment in photogrammetry software development, in which the analysis software interfaces directly with the measurement software, is also discussed.

3-1: Introduction to Videogrammetry

One of the primary advantages of ultra-lightweight and inflatable structures in space applications is the ability of these structures to transition from an initially small, packed volume to a large, deployed configuration. Understanding the deployment dynamics of ultra-lightweight and inflatable space structures is a key element of making them a reliable and robust technology. Because attaching traditional shape or vibration measurement devices to a deploying structure could significantly affect dynamic measurement, non-contacting optical measurements of the deployment process are desirable. Photogrammetry can be applied to measuring the geometry of these structures during deployment using a technique known as videogrammetry, which is essentially photogrammetry applied to a time sequence of images.

3-2: Overview of Poly tubing Structures

To evaluate videogrammetry application to deploying structures, inexpensive polyethylene tubes (“Poly tubing”) were used to make test articles. The Poly tubing

structures discussed here used black, 6-mil Poly tubing manufactured by Uline, Inc., which is commonly used in the packaging industry. Poly tubing offers several advantages as a deployment test article: it is flexible enough to be inflated and deflated; it is rugged enough to withstand multiple deployments; it comes in a variety of colors (transparent and black, in particular); and it is inexpensive and easy to work with.

Two types of deployable test articles were made of Poly tubing: single columns and tripods. The single-column test articles consisted of a length of Poly tubing (typically ranging from 36" to 60" long), which was heat sealed at one end and attached to a mounting plate at the other end. The mounting plate provided an air-hose connection as well as a stable base for the structure. Tripod test articles consisted of three single column articles joined at the heat sealed end. Each of the columns composing the tripod had a separate base plate and was inflated using a common air hose. A pressure regulator was installed between the high-pressure shop air supply and the test articles. The inflation pressure applied controlled the deployment speed of the tripods and columns.

3-3: Experimental Setup of VMD2Cam System

Videogrammetric measurements of deploying test articles were made using the Video Model Deformation Two Camera System (VMD2Cam) developed by High Technology Corporation. The system uses two cameras, a frame grabber, and a personal computer running the VMD2Cam software to track in real-time the three-dimensional locations of high-contrast targets. Originally designed to unobtrusively measure the wing deformation of wind-tunnel models at the NASA Langley Research Center, the system was here used successfully to measure the deployment of Poly tubing test articles.

The steps involved in making photogrammetric measurements (i.e., camera calibration, measurement planning, taking the photographs, importing the photographs into the photogrammetry software, target marking, target referencing, processing the data, and exporting the three-dimensional coordinate data) are also incorporated in the VMD2Cam system. The positions of the two cameras relative to the test article are chosen first, corresponding to the measurement planning stage. Once the cameras are in position, the cameras are calibrated by imaging the optical targets on a test article of known dimensions. The dimensions of the test article are copied into a file accessed by the VMD2cam software, and with this information the camera parameters and the camera locations and orientations are calculated. Obtaining the location and orientation of the cameras prior to measurement speeds the calculation of the locations of the targets during measurement, but also requires that the cameras be re-calibrated if the cameras are moved. The frame grabber simultaneously digitizes the analog video signals from each camera and imports the digital images into the software. Before the test article is deployed, the targets are marked and referenced in single photographs taken by each camera. During deployment, the software automatically tracks the targets from their initial locations, alleviating the need to mark and reference targets in every photograph in the time-series of images taken by the system. As the software tracks the locations of the targets on the deploying test article, the three-dimensional locations of those targets are calculated and exported to a text file. The text file contains the time-referenced locations of every marked target during deployment. Matlab was then used to visualize the exported data.

3-4: Single-Column Tests

The simplest type of test article used was a single column. The objective of these experiments was to develop a way to reliably track the locations of targets placed on the column as it deployed. Because the VMD2Cam system uses only two cameras, the targets on the column had to stay near a certain orientation, otherwise the targets would become hidden from one or both of the cameras and position data would be unobtainable. The resolution to this issue was obtained by investigating the methods of packing the columns to insure a measurable deployment.

3-5: Packing of the Single-Column Test Articles

Existing methods of packing inflatable/deployable tubes are the “Z-fold” and “roll” methods (Figure 10). In both of these methods, the tube is flattened prior to packing. In a Z-fold, the flattened tube is folded over itself repeatedly. In a roll, the tube is rolled up over itself.

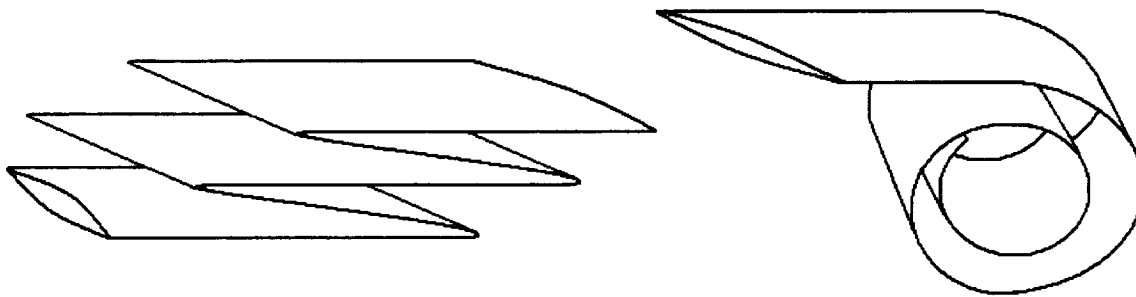


Figure 10: Examples of Z-fold and roll

Both of these packing methods were found to be inadequate during testing. The columns were intended to deploy vertically, extending in either an upward or downward direction. Neither method allowed a downward deployment because the columns would deploy on their own by the force of gravity and thus end the deployment tests before they began. Tube packing using both methods also failed to deploy upwardly in a reliable fashion, with partially inflated columns toppling over at various stages of deployment.

When the columns toppled, the attached targets moved out of view and position data became unobtainable. These failings were due to gravity, and both might be corrected with the use of Velcro between the packed layers of the column. Instead of pursuing modifications to these methods, however, a new packing method was developed that would perform in the desired fashion.

3-6: The Cuff-Fold

The Cuff-fold is an original method of packing deployable tubes developed during the course of this research. The Cuff-fold is a method of packing deployable tubes so that the deployment of the tube is more reliable and predictable than the other methods examined (Figure 11). The increased reliability and predictability of the Cuff-fold has two primary advantages. Because of these improvements, this method can be used in both experiments and applications with greater confidence. In addition, these improvements also simplify the planning and measurement stages of videogrammetric measurement of the deployment. Another practical advantage of the Cuff-fold is that the friction of the Cuff-fold packing method allows packed tubes to be suspended upside down without deploying unintentionally.

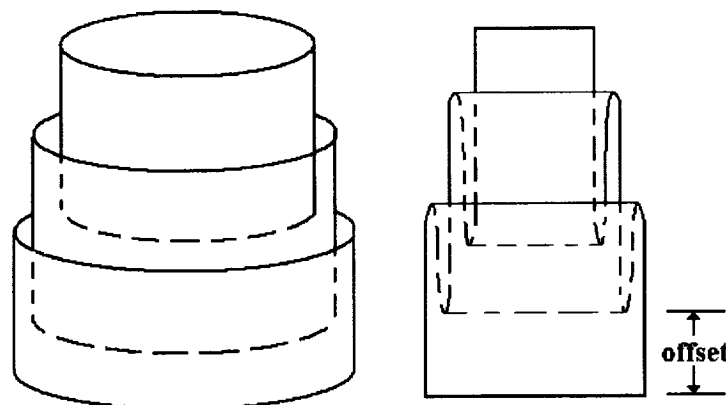


Figure 11: Example of Cuff-fold

The cuffs can be offset as shown in Figure 11, allowing retro-reflective targets to be placed on the exposed portion of each cuff and thus allowing a videogrammetric analysis of the deployment. Alternatively, the cuffs can be completely nested, in which case only the outer-most cuff would be visible initially. This second approach is more difficult for photogrammetric analysis, but the packaged volume is smaller and thus more practical in space applications. Experimental analysis of Cuff-folded tubes with offsets should allow general analytical models of Cuff-folded tubes to be validated, which could, in turn, be used to predict the dynamics of Cuff-folded tubes that are completely nested. When deploying, Cuff-folded tubes extend in a nearly telescopic fashion. The deployment of Cuff-folded tubes is linear and orderly, which is advantageous for both measuring the tube during deployment and for evaluating the potential applications of deployable tubes in space structures.

Deployment measurement of Cuff-folded columns proved to be the most successful of all single column deployment tests. Although the Cuff-folded columns did tend to topple when deploying upwardly, downward deployment tests were successful in yielding position data of targets on the column during deployment. Figure 12 shows the three-dimensional coordinates of targets placed along the column during various stages of deployment. The data corresponding to the initial configuration of the column is shown in the upper-left portion of Figure 12, where the column is suspended vertically and packed using a Cuff-fold with targets placed on the offsets. A horizontal set of four reference targets is also visible above the column. The “current frame” above each image indicates the location of the data in the time sequence, which covers 176 individual three-dimensional measurements. It can be seen that as time progresses, the column deploys

downward in a reasonably linear fashion until the final, deployed configuration is achieved in the lower-right portion of Figure 12.

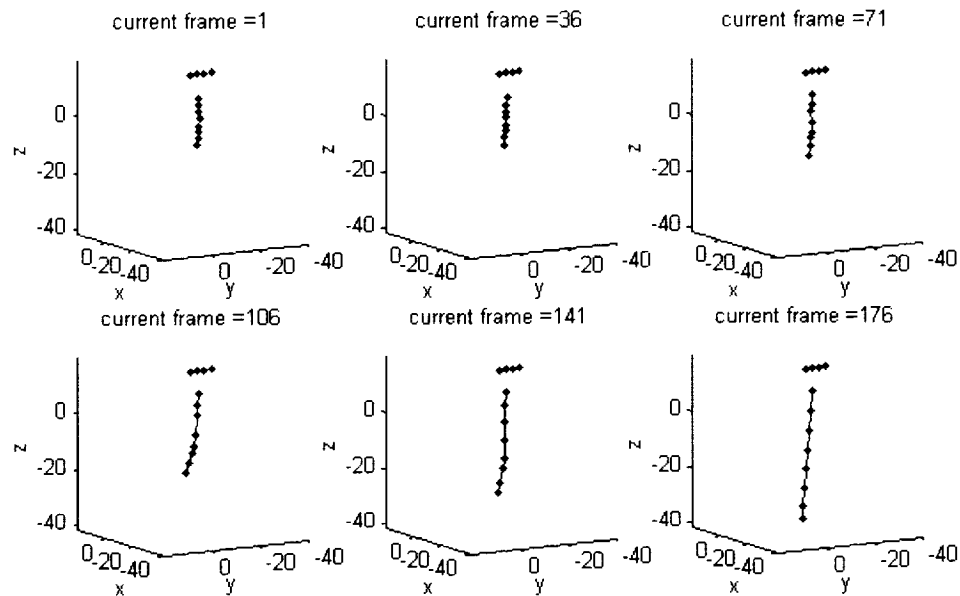


Figure 12: Downward deployment tests data from a Cuff-folded column. The four static points near the top of the column are a reference attached to the supporting structure.

3-7: Tripod Tests

To help understand the issues involved with the deployment of complex structures consisting of multiple components, the inflation of Poly tubing tripods was also studied. The goal of the experiments was to use the VMD2Cam system to gather position data from targets placed on two or three legs of the tripod as it deployed. It became obvious early in the tests that measuring targets on all three legs would be prohibitively difficult using the two-camera VMD2Cam system.

Two configurations were used in attempts to measure all three legs (Figure 13). In each configuration, one leg was centered in the field of view of both cameras with the other two legs visible near the edges of each view. In Configuration A, the center leg was placed toward the cameras. This configuration failed because the center leg was found to block the other legs from the view of each camera during deployment. In Configuration

B, the center leg was placed away from the cameras. This configuration also failed because the view of the center leg was blocked by the other two legs during deployment. Surrounding the tripod with cameras would have enabled at least two cameras to view each leg during deployment. However this was not possible using the VMD2Cam system. The tripod tests were therefore constrained to measure only two of the legs at a time during deployment in a configuration similar to Configuration B.

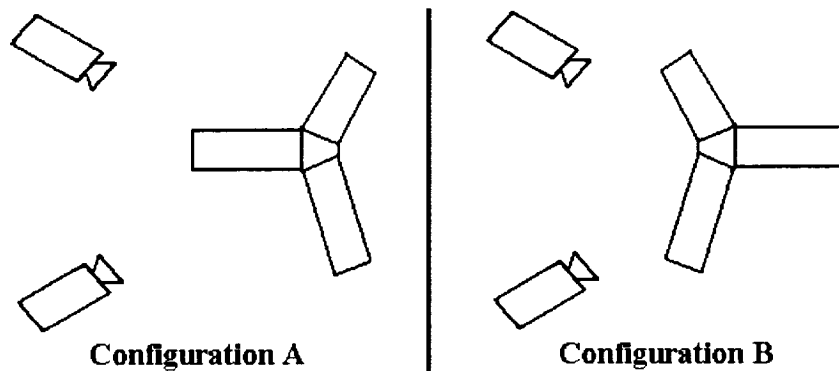


Figure 13: Configurations used in attempts to measure all three tripod legs

Even during the measurement of only two legs, targets were frequently lost from the view of either or both cameras. Typically, this was due to the orientation of the targets relative to the cameras, and the problem could likely be eliminated by using a videogrammetry system with more cameras. Another difficulty encountered during these tests was a more fundamental problem involving complex deployable structures. As the tripod deployed, it was possible for two or more legs to move into a configuration from which full deployment became impossible. This situation was termed “locking.” Locking could be induced by the way the structure was packed (Figure 14). Locking occurred because two or more of the legs would push in opposite directions at the vertex, which constrained further deployment. Figure 14 shows the folding approach used to avoid locking of the tubes.

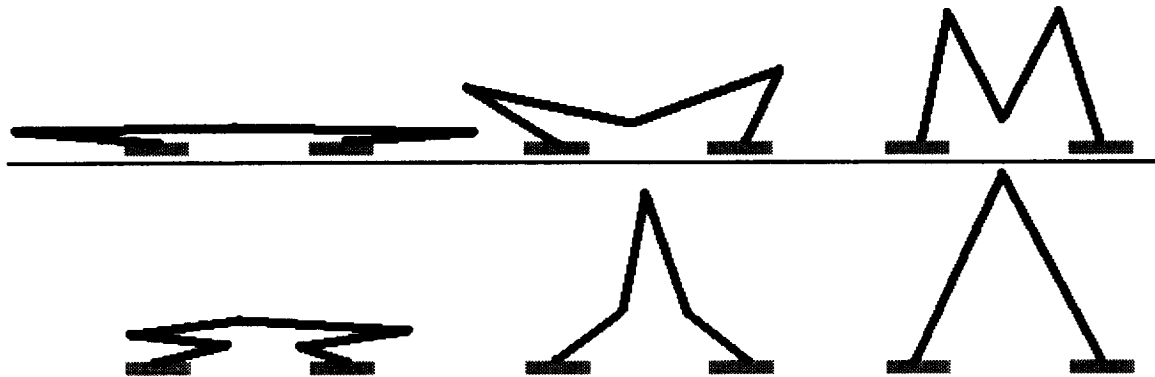


Figure 14: The top series illustrates the front two legs of a tripod transitioning from a packed state to a “locked” state; the bottom series illustrates the front two legs of a tripod transitioning from a different packed state to a successfully deployed state

3-8: Retro-Reflective vs. Flat White Targets

During deployment, the geometry of the structure will change significantly and any targets attached to the structure will likely pass through a variety of orientations relative to the cameras. In order for tracking software to be effective, the targets must remain as visible as possible during the entire deployment process. Both retro-reflective and flat white targets were used during the VMD2Cam series of videogrammetry measurements of deploying structures, with both types of targets proving useful but without a conclusive “best choice.”

A series of tests were conducted to understand how the visibility of retro-reflective and flat white targets change as the angle between the camera and the normal to the targets increase (Figure 15). Retro targets are very reflective, particularly with flash illumination, and flat white targets are similar to white paper. Both types of targets were punched from sheets of the material, and pairs of retro and flat white targets were attached to a support (Figure 15). Four sets of target pairs were placed at varying angles (0°, 45°, 60°, 75°) relative to the camera and photographed under varying lighting

conditions. The targets were illuminated by a variable light source placed as close to the camera as possible.

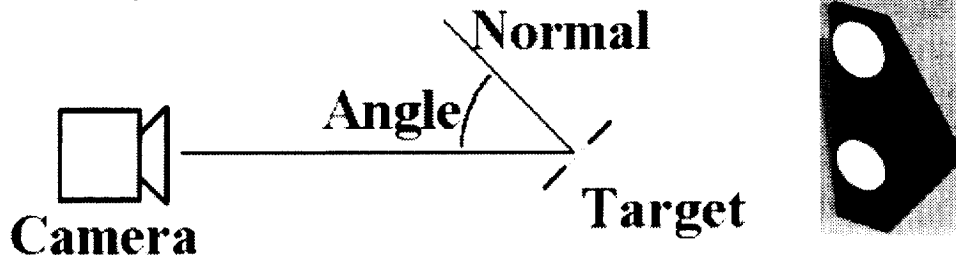


Figure 15: Experimental setup of the target comparison tests
Left- The relation between the camera and the normal of the target.
Right- A photo of a pair of targets attached to a support.

A Matlab function was written to analyze the resulting variation in visibility as a function of angle. Figure 16 shows the results of this analysis. Amplitude refers to the peak intensity of the target as imaged by a DC290 camera, and ranges from 0 to 255. Retro-reflective targets are more visible than flat targets at a given light level at most angles. However, the visibility of retro targets drops off much faster than flat white targets at high angles of incidence. Because flat white targets have a more-constant visibility over a wider range of angles, these targets are probably the better choice for videogrammetry involving deploying structures that will drastically change geometry. For measurements involving less dramatic changes in geometry, retro-reflective targets may be preferred due to the increased visibility at lower light levels and generally higher contrast with the surrounding structure.

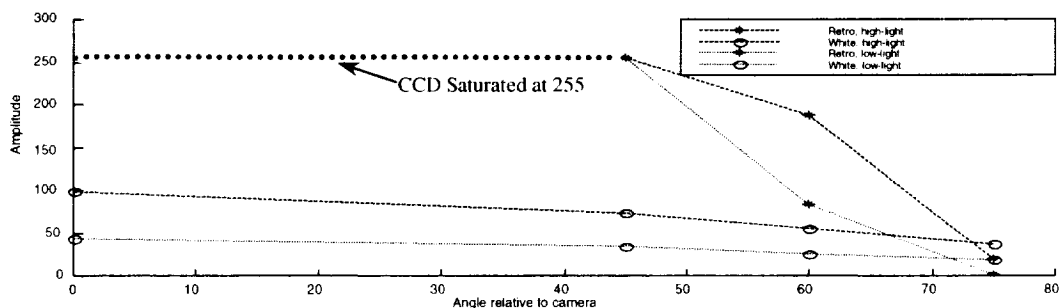


Figure 16: Variation in amplitude as a function of angle for various targets and lighting conditions

3-9: Dot Projection and Membrane Materials

One promising alternative to using physical targets for both static and dynamic shape analysis is dot projection, in which dots of white light are projected onto a surface and used as targets for the purpose of photogrammetry (Figure 17). The benefits of dot projection are twofold: dot projection allows for thousands of targets to be quickly and easily distributed on a surface, and dot projection eliminates the effects of placing physical targets on a surface, which add undesirable mass or stiffness to an ultra-light structure. These advantages make dot projection an attractive option for ultra-lightweight membrane structures.

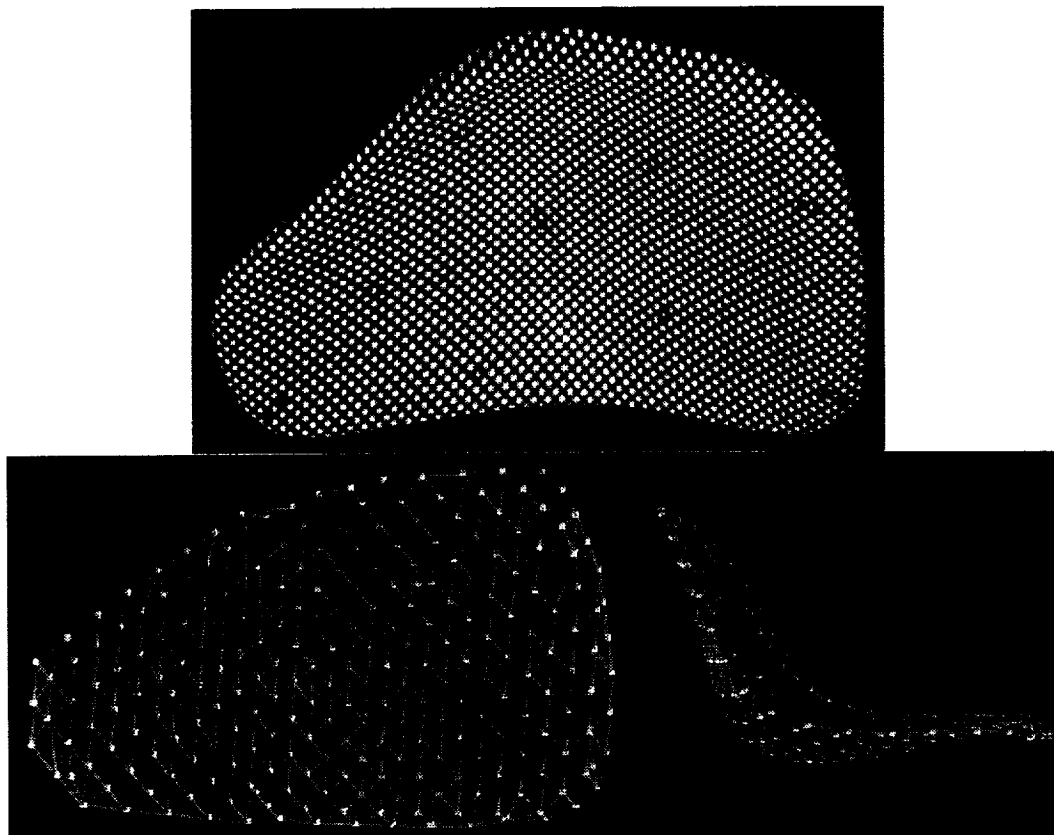


Figure 17: Dot projection (above) and resulting wire frame data (below) of a chair

It is important to note that when using dot projection with videogrammetry, the projected targets on the structure will not move with the structure but will instead move

as a result of the changing shape of the structure: the motion of a specific point on a structure cannot be tracked using dot projection. This makes dot projection inappropriate for vibration analysis. However, a time series of measurements can be made of the structure as a whole using dot projection as long as it is understood that the targets do not represent specific points on the structure. For example, the parabolic shape of a concentrator similar to the 5m concentrator could be measured using dot projection. If the internal pressure of the concentrator was varied with time, dot projection could be used to measure the changing shape of the concentrator (i.e. the changing focal length) as a function of time even though the trajectories of specific points on the concentrator could not be obtained using this method.

To better understand which types of materials are most amenable to measurement by dot projection photogrammetry, a variety of materials were examined. The material properties of relevance to dot projection are reflectivity, absorptivity, and transparency. Reflectivity is of two types: specular reflection and diffuse reflection. Specular reflection occurs on surfaces that are smooth on a microscopic level, whereas diffuse reflection occurs on surfaces that are microscopically rough (Figure 18). With specular reflection, all of the light reflects in one direction and the surface behaves like a mirror. Therefore, the projected dots will be most visible in only one direction. With diffuse reflection, the light reflects in many directions, and projected dots will be visible from many locations. Because photo/videogrammetry uses multiple cameras in multiple locations, diffuse reflection is much preferred over specular reflection.

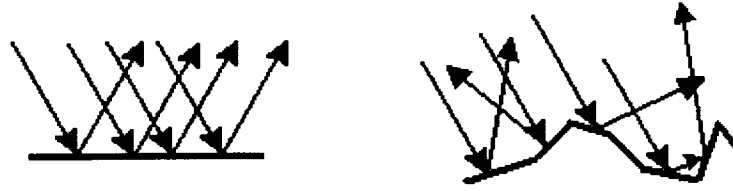


Figure 18: A microscopic perspective of how light reflects off a specular (left) and diffuse (right) surface

Absorptivity and transparency are related in that both properties cause a loss of light reflected off a material. Darkly colored materials absorb more light than lightly colored materials, and materials that are more absorptive will be more difficult to measure using dot projection since the dots will appear dim on the structure. Transparent materials allow most of the light to pass through the material, and so highly transparent materials will also be difficult to measure using dot projection.

This qualitative discussion leads to a recommendation for a diffuse, low-absorption, low-transparency material. To validate this recommendation, a series of experiments was carried out on various materials. Table 7 lists the materials that were examined and a qualitative description of their material properties. The opaque materials were attached to a vertical backing and the transparent materials were attached to supports at only the top and bottom. Dots were projected onto the materials and photographed under four conditions (Figure 19).

Table 7: Qualitative description of material properties

Material	Reflection	Transmission	Absorption
White Paper	diffuse	low	low
Kapton 500 MTB (black)	diffuse	low	high
Kapton 100 CB (black)	specular	low	high
Metalized Kapton (silver)	specular	low	low
Kapton HN 100 G (gold)	specular	low	medium
Semi-transparent Mylar (gray)	specular	high	low
Semi-transparent Kapton (orange)	specular	high	low

- Case 1: Materials, camera, and projector are aligned, with ambient lights on
Case 2: Materials, camera, and projector are aligned, with ambient lights off
Case 3: Materials and projector are aligned, with the camera at an angle and ambient lights off
Case 4: Camera and projector are aligned, with the materials at an angle and ambient lights off

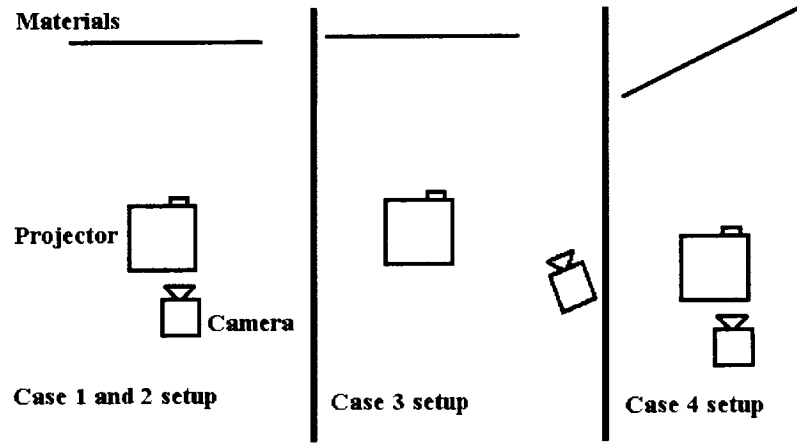


Figure 19: Experimental setup of the dot projection cases

An attempt was made to measure quantitatively contrast. Contrast was defined as the difference in intensity between the brightest part of the projected dot and the darkest part of the area surrounding it. This value can vary between different areas of the material, and so the maximum and minimum contrasts are given (Table 8).

Table 8: Quantitative contrast measurements

Material	Case 1 Contrast	Case 2 Contrast	Case 3 Contrast	Case 4 Contrast
White Paper	120, 107	139, 125	153, 146	129, 106
Kapton 500 MTB (black)	158, 51	143, 86	Not visible	Not visible
Kapton 100 CB (black)	199, 46	199, 45	Not visible	Not visible
Metalized Kapton (silver)	195*, 108	218, 48	208, 31	Not visible
Kapton HN 100 G (gold)	183*, 57	182*, 131	230, 178	Not visible
Semi-transparent Mylar (gray)	Not visible	Not visible	175, 50	Not visible
Semi-transparent Kapton (orange)	Not visible	Not visible	240*, 89	Not visible

* Indicates that the CCD was saturated to the maximum intensity level of 255 in the region of a projected dot, and therefore the actual contrast may be higher than that measured here.

It is desirable to have consistently high contrast levels with dots that are visible in all the configurations examined. The black Kapton materials provided a highly variable contrast level and were not visible in two of the configurations. The Metalized Kapton and Kapton HN, both of which had Metalized surfaces, provided high contrasts but also

had a tendency to generate hotspots (regions in which the image is overwhelmed with reflected light) and were not visible in one of the configurations. The semi-transparent materials also produced hotspots and were not visible in three of the configurations. The white paper provided the best overall visibility of the projected dots, which is consistent with the earlier hypothesis that a diffuse, low-transmission, and low-absorption material would be best.

3-10: Matlab/PhotoModeler DDE System

The two primary pieces of software used throughout this thesis research are Matlab and PhotoModeler. Matlab is a powerful technical computing environment, and PhotoModeler is a capable and easy to use photogrammetry software package. The strengths of these two separate software systems can be effectively combined using Dynamic Data Exchange (DDE). Dynamic Data Exchange allows separate computer programs to communicate in real time, and commands can be issued by one program and executed by another. This allows Matlab to use PhotoModeler as a “photogrammetry engine.” Matlab can use PhotoModeler to handle all of the complex photogrammetry calculations, and thus custom photogrammetry and videogrammetry software can be written in Matlab without having to recreate the involved programming of a completely new photogrammetry system. Matlab already has the basic functions required to manipulate images, and Matlab functions can be edited and expanded at anytime. DDE uses a library of functions determined by PhotoModeler, and that DDE library is already large and expanding.

A DDE session is orchestrated as follows. Both Matlab and PhotoModeler must be running. Matlab initiates a DDE session using built-in functions and a PhotoModeler-

specific address. Once the session is initiated, Matlab can command PhotoModeler to execute a number of commands, and PhotoModeler can return information to Matlab such as data and error indicators. When the session is over, Matlab terminates the data exchange.

A prototype photogrammetry system was successfully written in Matlab using the Matlab/PhotoModeler DDE capabilities (see “ddedemo.m,” p. A-6 in the Appendix). In this system, Matlab initiates a DDE session with PhotoModeler and commands PhotoModeler to mark and reference a series of target locations (two-dimensional points in a series of photographs) in a blank project. When all of the targets are marked and referenced, Matlab commands PhotoModeler to process the data. Once the data is processed, PhotoModeler returns the three-dimensional locations of the targets to Matlab, and the session terminates. Thus, Matlab is able to use PhotoModeler to photogrammetrically calculate the three-dimensional locations of targets. All that remains is to write a set of Matlab functions to handle sub-pixel target marking, target tracking, and image sequencing, and a Matlab/PhotoModeler-based videogrammetry system would be available, with all of the accuracy demonstrated by PhotoModeler and all of the power and flexibility available with Matlab (Figure 20).

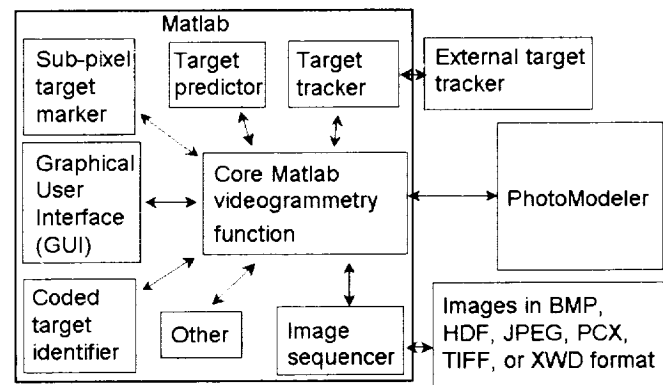


Figure 20: A possible layout for a full Matlab/PhotoModeler-based videogrammetry system

Chapter 4 : Recommendations for Future Work

In this chapter, areas of potentially productive future research are suggested based on the results of the research conducted. Hardware issues and software development are examined, as are the promise of target coding and inflation control.

4-1: Increased Number of Cameras and Software Development

Future videogrammetric tests of deployable structures will benefit by using systems that use more than two cameras. There are three primary advantages to using such a system as opposed to a two-camera system: increased accuracy, increased precision estimation, and increased visibility of the test article during deployment. As the number of cameras viewing a target increases, the ratio of the number of equations to the number of unknowns in the bundle adjustment increases, improving accuracy. Precision estimates of the measurement can be generated by calculating the difference between the marked location of a target on a photograph and the projection of the calculated position of that target onto the photograph. If the same target appears in a sufficient number of photographs, the software can remove the camera views of that target having the largest discrepancy between measured point location and calculated point location, which will reject outliers in the solution calculation, which further improves both accuracy and precision.

During the deployment process, targets visible to a particular camera at one stage of the deployment may become hidden from the view of that camera at a later stage due to the changing geometry of the test article. Conversely, targets that were once hidden from a particular camera may become visible. To effectively use a multi-camera system, targets that move into the view of a camera should be recognized, marked, and tracked to

increase the accuracy of the photogrammetric solution. Also, the ability to track a previously hidden target will negate the negative effects of losing the view of that target from another camera.

When a target is visible from at least two cameras, it is possible to determine the three-dimensional position of that target. Given that the orientation and location of all the cameras is also known at this stage of photogrammetric measurement, it should be straightforward to project the location of the target onto the image planes of the cameras not already tracking that target. A target-marking algorithm could then begin to search for the previously hidden target in that area to determine if it has become visible. Consideration should be given to the case of multiple targets with distinct positions in space being projected onto the image plane in close proximity, thus creating a situation where the targets might be misidentified using this process.

If the positions of a target cannot be determined due to a loss of target visibility, there are at least two methods to relocate that target using other cameras. The first is analyzing the trajectory of the target to predict a likely location of the target, and the second is using known constraints of the test article to locate the target position relative to other targets with known positions. The first method is described in Reference 12, where a process uses the past locations of a target with a Lagrange Polynomial to predict the likely subsequent location of that target. This location can then be used to begin a search for the target in the image planes of various cameras.

The second method reduces the possible volume of space in which the target is located using the known locations of other targets on the structure. For example, consider a deploying tube one foot long and with three targets spaced every six inches. If the

locations of the top and bottom targets are known, then the location of the middle target is constrained to be within the intersection of two spheres centered at the two known targets and with radii of six inches. That is, the middle target cannot be located more than six inches from either the top or the bottom target. This reduces the space that must be searched to find the middle target in other camera views.

4-2: Coded Targets

Another way to identify newly visible targets is to use coded targets. Coded targets surround a central target with a set of visual markers: the central target is used in sub-pixel marking, and the perimeter code is used for automatic referencing. Coded targets are already implemented in many photogrammetric and videogrammetric systems, including VSTARS. The concept has been proven useful for many applications. However, a difficulty arises when considering the use of coded targets on deploying structures.

If the central target of a coded target is to be the same size as a non-coded target (thereby giving both targets the same sub-pixel marking accuracy), the coded target must be physically larger than the non-coded target in order to accommodate the perimeter code. That means that more of the target must stay in the view of the cameras, especially considering that both the central target and the perimeter code must remain visible and distinct. This will restrict the measurable orientations of the target, but this problem might be alleviated by using several cameras spaced around the test article. It may also be advantageous to use a mixture of coded and non-coded targets, with the coded targets acting as “landmarks” for the photo/videogrammetry software, and the non-coded targets providing the majority of the three dimensional data.

4-3: Inflation Control

The two largest problems encountered while conducting test measurements of tripod deployment were loss of target visibility and locking. In the far term, it should be possible to integrate real-time videogrammetric measurement and deployment control of deployable structures. This would allow for active prevention of both locking and loss of target visibility during deployment. For example, if the system detected a leg of a tripod passing near the edge of visibility, it could reduce the inflation speed of the other legs to draw the structure back into a better orientation. If the structure approached a locked configuration, the system could reduce pressure in one or more legs before completing the deployment. Such a system would also be of value for the deployment of inflatable structures in space, as it would simultaneously improve data collection of the deployment and act to keep the structure safe from damage during deployment.

Chapter 5 : Concluding Remarks

Photogrammetry is a highly appropriate measurement technique for studying the static shape of ultra-lightweight and inflatable structures, and videogrammetry is equally well suited for studying the dynamics of these structures. However, it appears that the practical issues involved with measuring particular ultra-lightweight and inflatable structures will be problem-specific, and so a general-purpose photo/videogrammetry system will be most useful if it is highly flexible. Consideration must be given to both hardware and software flexibility and accuracy.

It has been shown that consumer cameras with resolutions of 2-3 megapixels yield good precisions in photogrammetric measurements. Professional cameras with resolutions of 4-6 megapixels cost significantly more but should be expected to yield higher precision. The advantages of consumer cameras come from lower cost and faster product development. Because consumer cameras are less expensive, a larger number can be purchased for measurements. Fast product development is narrowing the gap in resolution between consumer and professional cameras, and because consumer cameras cost much less, it is more feasible to upgrade them when new technologies are brought to market.

For static measurements in which the test article is rigid and in a suitable environment, a high-resolution, professional camera could image it from several locations by a photographer moving around the test article, yielding maximum accuracy. However, if the structure is not rigid (e.g., a membrane), multiple cameras may be required to image the article simultaneously from all necessary camera stations. In addition, if the structure is in an environment requiring remote operation of the cameras (e.g. a vacuum

chamber), multiple cameras will be required. Because both of these scenarios are likely to occur with ultra-lightweight and inflatable structures, multiple cameras will generally be required. For this purpose, a large number of consumer cameras may be most appropriate. However, the present gap in resolution between consumer and professional digital video cameras (0.3 vs. 1 megapixel) is large enough to require professional camera systems for high-precision measurements. Experiments involving the two video camera VMD2Cam videogrammetry system have illustrated the need for at least three or four video cameras for use in videogrammetry measurements of test articles that will undergo large changes in geometry. Digital video systems are capable of taking static images. Therefore, the optimal photo/videogrammetry system might incorporate a small number of professional digital video cameras for use in both static and dynamic measurements, as well as a number of supplemental consumer cameras for increased accuracy in static measurements. It is worth noting that PhotoModeler is capable of using images from cameras with varying resolutions in the same measurement.

It has been shown that materials have a significant effect on how well measurements can be made using both photogrammetry and videogrammetry. When choosing between flat white and retro-reflective adhesive targets, consideration should be given to the lighting conditions and the expected orientations of the targets relative to the cameras. While dot projection techniques offer several advantages for measuring the shape of ultra-lightweight and inflatable space structures (e.g. tens of thousands of targets are almost instantly available for measurement without adding mass or stiffness to the structure), the materials used to make these structures are typically reflective or transparent. When using dot projection, materials like these are much more difficult to

measure than materials that are opaque, light colored, and diffuse, and so the materials used to create test articles may be dictated by how they are to be measured.

A Matlab/PhotoModeler-based photo/videogrammetry system will likely yield the accuracy and flexibility required for an effective ultra-lightweight and inflatable space structure measurement system. The limiting factor in the accuracy of PhotoModeler appears to be the camera calibration technique. This could be improved with either more-refined and specialized calibration software or by determining the camera parameters in an optical laboratory. The DDE interface capability of Matlab and PhotoModeler should contribute significantly to the flexibility of the photo/videogrammetry system software, as should the likelihood of increased DDE capabilities in future versions of the PhotoModeler software.

In a photo/videogrammetry system, the hardware and software must complement each other to be effective, and the same is true for the measurement system as a whole and the test article being measured. Test articles will need certain optical properties if they are to be effectively measured using either photogrammetry or videogrammetry. As more experience is gained making these types of measurements, both the test articles and the measurement systems will be improved and refined. Eventually this process will yield the validation necessary to capitalize on the immense potential of ultra-lightweight and inflatable structures in space-based applications.

References

- [1] "The Gossamer Spacecraft Initiative: Technology to See the Unseen Universe," First Gossamer Workshop, Oxnard, CA, October 1999.

- [2] Chmielewski, A. B., Moore, C., and Howard, R., "The Gossamer Initiative," IEEE paper 0-7803-5846-5/00, January 2000.

- [3] First Gossamer Spacecraft Forum, Held in Conjunction With the AIAA Structures, Structural Dynamics, and Materials Conference, Atlanta, GA, April 2000.

- [4] Wolf, P. R. and Bon, A. D., *Elements of Photogrammetry With Applications in GIS*, 3rd edition, McGraw-Hill, Boston, MA, 2000.

- [5] Church, E. F., *Elements of Photogrammetry*, Syracuse University Press, Syracuse, NY 1948

- [6] Karara, H. M., *Handbook of Non-Topographic Photogrammetry*, 2nd edition, American Society of Photogrammetry, Falls Church, VA, 1989.

- [7] Granshaw, S. I., "Bundle Adjustment Methods in Engineering Photogrammetry," *Photogrammetric Record*, Vol. 10, No. 56, 1980, pp. 181-207.

- [8] DiChristina, M., "Solar Booster," *Popular Science Magazine*, October 2000, p. 28

- [9] Dornhein, M., "Inflatable Structures Taking to Flight," *Aviation Week & Space Technology*, January 25, 1999, pp. 60-62.
- [10] Pappa, R. S., Giersch, L. R., and Quagliaroli, J. M., "Photogrammetry of a 5m Inflatable Space Antenna With Consumer Digital Cameras," Proceedings of the 19th International Modal Analysis Conference, Kissimmee, FL, February 5-8, 2001
- [11] *PhotoModeler Pr, Version 4.0d User Guide*, Eos Systems Inc., 1993 – 2001.
- [12] Fraser, C. S. and Shao, J., "On the Tracking of Object Points in a Photogrammetric Motion Capture System," Proceedings of SPIE Vol. 4309, 2001.

Appendix

All of the codes listed here were written by Louis Giersch during the course of the research described in the preceding chapters. This is not a complete list of every code written, but rather only the codes thought most important to the thesis. These codes may be reproduced and used as long as the following requirements are met:

- 1) Credit is given to the original author
- 2) It is understood that while no errors are known to exist in these codes, they are by no means guaranteed. The original author is by no means liable for any damages resulting from the use of these codes

Digita script used for DC290 settings

```
name "startup"
mode 0
menu "Kodak Scripts"
label "startup"

declare u: uSelect      ###Set up variables
declare i: iFlagOn, iFlagOff  ###Set up variables
SetCameraState ("zpos", 100)  ###Set Zoom, Min = 100, Max = 300
DisplayLine ("Zoom: Fully Out")
SetCameraState ("fmod", 3)    ###Specifies focus mode, 3 = Manual
SetCameraState ("fdst", 250)  ###Set focus distance, 50 = Min
DisplayLine ("Focus Distance: 5m")
SetCameraState ("smod", 2)    ### 2=Fill -Built-in flash fires in all situations.

TheEnd:
Cancel:
Quit:
Exitscript
```

Matlab code "opendata2.m" used to read xyz and precision data exported from PhotoModeler

```
function Pres_Pts = opendata2(filename)
%function Pres_Pts = opendata2(filename)
%Written by Louis Giersch, Sept. 15, 2000
%
%This function opens the file 'filename', a .txt file exported by
%PhotoModeler, and reads the data into an array called Pres_Pts which
```

```

%contains the precision of x,y, and z locations as well as the x,y, and
z locations.
%
%filename is a string, for example: 'table1.txt'. The file must
%be located in a directory referenced as a path by Matlab.

fid = fopen(filename, 'r');          %open file in binary mode.
n=1;                                %n is an index.
line = fgetl(fid);                   %skip first line.
while feof(fid) == 0 %stop while loop when end of file (EOF) is found.
    line = fgetl(fid);               %get a (string) line from the file.
    spaces = isspace(line);           %find the spaces in the string.
    numbers = find(~spaces);          %find the non-spaces in the string.
    num1 = numbers(1); %the first number starts at the first non-space.
    x1 = 1;                          %starting at the beginning of the string.
    while (numbers(x1+1) - numbers(x1) == 1) %find a jump in non-space
                                                %locations.
        x1 = x1 + 1; %move through the string.
    end                                %end search when jump is found.
    x1 = x1 + 1;
    num2 = numbers(x1); %this is where the second number on the line is.
    while (numbers(x1+1) - numbers(x1) == 1) %search for another jump in
        x1 = x1 + 1;                    %non-space locations.
    end
    x1 = x1 + 1;
    num3 = numbers(x1);
    while (numbers(x1+1) - numbers(x1) == 1) %search for another jump in
        x1 = x1 + 1;                    %non-space locations.
    end
    x1 = x1 + 1;
    num4 = numbers(x1); %this is where the fourth number on the line is.
    while (numbers(x1+1) - numbers(x1) == 1) %search for another jump in
        x1 = x1 + 1;                    %non-space locations.
    end
    x1 = x1 + 1;
    num5 = numbers(x1);
    while (numbers(x1+1) - numbers(x1) == 1) %search for another jump in
        x1 = x1 + 1;                    %non-space locations.
    end
    x1 = x1 + 1;
    num6 = numbers(x1);
    s1 = line(num1:num2-1);            %
    s2 = line(num2:num3-1);            %get the sub strings containing
    s3 = line(num3:num4-1);            %the three numbers.
    s4 = line(num4:num5-1);            %
    s5 = line(num5:num6-1);            %
    s6 = line(num6:length(line));       %
    Pres_Pts(n,1) = str2num(s1);        %
    Pres_Pts(n,2) = str2num(s2);        %convert from strings to actual
    Pres_Pts(n,3) = str2num(s3);        %numbers, put into POINTS.
    Pres_Pts(n,4) = str2num(s4);        %
    Pres_Pts(n,5) = str2num(s5);        %
    Pres_Pts(n,6) = str2num(s6);        %
    n = n + 1;                          %go to the next line.
end
fclose(fid); %close file

```

Matlab function "fitparabola.m" used to fit xyz data to a parabolic surface

```
function d2 = fitparabola(d, maxitr, graphics)
%function fitparabola(d, maxitr, graphics)
%Written by Louis Giersch
%Fits the input to the function  $z = a*x^2 + b*y^2 + c*x + d*y + e$ ,
%which is an axially symmetric paraboloid
%x,y,z must be column vectors
%graphics = 0,1,2
%This function attempts to align the axis of symmetry with the z axis
x = d(:,1);
y = d(:,2);
z = d(:,3);
num_of_pts = length(x);
if(num_of_pts < 4)
    error('There must be at least 4 points to fit the paraboloid')
end
xi = x;
yi = y;
zi = z;
%The axis of symmetry is aligned with the z-axis by rotating about the
%x-axis by an angle a1 and -a1, and about the y-axis by an angle a1 and
%-a1. This results in four new data sets, all of which are then curve
%fit to the equation:  $z = a*x^2 + b*y^2 + c*x + d*y + e$ . The data set*
%with the lowest resulting RMS is the assumed to be the set most
%aligned with the z axis. This set is again rotated by a reduced angle
%value angred*a1 (angred < 1), and this continues for maxitr
%iterations. If angred is not < 1, the process will not converge.
itr = 1;
a1 = 10*2*pi/360;           %initial rotation angle
angred = .95;               %reduce a1 by this much after each iteration
showme = zeros(1, maxitr);
while (itr <= maxitr)        %% Begin alignment iterations
    vec = [x';y';z'];
    sa1 = -sin(a1);          % calculate sin, cos of a1 here to speed
    ca1 = cos(a1);           % execution
    msal = -sin(-a1);        %
    mcal = cos(-a1);         %
    R1 = [1 0 0; 0 ca1 sa1; 0 -sa1 ca1]; % The four rotation
    R2 = [1 0 0; 0 mcal msal; 0 -msal mcal]; % matrices used
    R3 = [ca1 0 sa1; 0 1 0; -sa1 0 ca1]; %
    R4 = [mcal 0 msal; 0 1 0; -msal 0 mcal]; %
    vec1 = (R1*vec)'; % Rotate about x axis by a1
    x1 = vec1(:,1); % Results are Set 1
    y1 = vec1(:,2); %
    z1 = vec1(:,3); %

    vec2 = (R2*vec)'; % Rotate about x axis by -a1
    x2 = vec2(:,1); % Results are Set 2
    y2 = vec2(:,2); %
    z2 = vec2(:,3); %

    vec3 = (R3*vec)'; % Rotate about y axis by a1
    x3 = vec3(:,1); % Results are Set 3
```

```

y3 = vec3(:,2); %
z3 = vec3(:,3); %

vec4 = (R4*vec)'; % Rotate about y axis by -a1
x4 = vec4(:,1); % Results are Set 4
y4 = vec4(:,2); %
z4 = vec4(:,3); %

index2 = 1;
while (index2 <= 4) % Start surface fit
    if(index2 == 1) % This section calculates the surface parameters
        xc = x1; % (including RMS error) for all of the data sets (1-
        yc = y1; % 4), starting with Set 1 (e.g. on the first pass
        zc = z1; % through, the current set [xc, yc, zc] is Set 1,
    end % on the second pass the current set is Set 2,
    if(index2 == 2) % etc.). index2 indicates the current pass.
        xc = x2;
        yc = y2;
        zc = z2;
    end
    if(index2 == 3)
        xc = x3;
        yc = y3;
        zc = z3;
    end
    if(index2 == 4)
        xc = x4;
        yc = y4;
        zc = z4;
    end
    end

A = zeros(num_of_pts,5); % This section does the
A(:,1) = xc.^2; % actual surface fitting and
A(:,2) = yc.^2; % determines the focal
A(:,3) = xc; % lengths and the RMS. The RMS
A(:,4) = yc; % of each set are stored in
A(:,5) = ones(num_of_pts,1); % the vector er, so that the
b = zc; % set with the smallest RMS
fit_vector = zeros(5,1); % error can be found once all
fit_vector = inv(A'*A)*A'*b; % of the sets have been
a = fit_vector(1); % analyzed.
b = fit_vector(2); %
c = fit_vector(3); %
d = fit_vector(4); %
e = fit_vector(5); %
focal_length_x1 = 1/(4*a); %
focal_length_y1 = 1/(4*b); %
zp = a*xc.^2 + b*yc.^2 + c*xc + d*yc + e; %
diff = zc - zp; %
max(diff); %
er(index2) = sum(diff.^2); %
index2 = index2 + 1; %
end % End Surface fit

if(find(min(er)==er) == 1) % This section picks out the set with
    x = x1; % the smallest RMS error (min(er)), and
    y = y1; % sets up the next iteration to use that

```



```

        z = z1; % set as the starting point for the
        showme(itr) = 1; % rotations.
    end
    if(find(min(er)==er) == 2) %
        x = x2; %
        y = y2; %
        z = z2; %
        showme(itr) = 2; %
    end
    if(find(min(er)==er) == 3) %
        x = x3; %
        y = y3; %
        z = z3; %
        showme(itr) = 3; %
    end
    if(find(min(er)==er) == 4) %
        x = x4; %
        y = y4; %
        z = z4; %
        showme(itr) = 4; %
    end

    end

    a1 = a1*angred; % Reduced the size of a1
    itr = itr + 1; % go to the next alignment iteration
    end % End alignment iterations
    %After the alignment iterations are complete, the characteristics of
    %the resulting data are determined
    focal_length = abs(focal_length_x1 + focal_length_y1)/2
    focal_pm = abs(focal_length - abs(focal_length_x1))
    zp = a*xc.^2 + b*yc.^2 + c*xc + d*yc + e;
    diff = zc - zp;
    average_error = mean(diff)
    sum_square_error = sum(diff.^2);
    RMS = norm(diff)/sqrt(length(diff)) %root-mean-square (RMS)
    err = diff.^2;
    aver = 1.1*sum_square_error/length(x);
    d2x = x(find(err < aver));
    d2y = y(find(err < aver));
    d2z = z(find(err < aver));
    d2 = [d2x'; d2y'; d2z'];

    %The rest of the code is graphics
    figure(4);
    plot([1:1:maxitr], showme);
    x2 = min(x):(max(x) - min(x))/50:max(x);
    y2 = min(y):(max(y) - min(y))/50:max(y);
    z2 = zeros(length(x2), length(x2));
    n = 1:1:length(x2);
    m = 1;
    while m <= length(x2)
        z2(m,n) = a*x2(m).^2 + b*y2(n).^2 + c*x2(m) + d*y2(n) + e;
        m = m + 1;
    end
    if(graphics >= 1)
        close all;
        figure(1);
        circ(x,y, diff*30);
    end

```

```

        xlabel('radii are proportional to the difference between expected
and actual z values (out of plane)');
        ylabel(strcat('average error = ', num2str(average_error), ', RMS = ',
num2str(RMS)));
        title(strvcat('z = a*x^2 + b*y^2 + c*x + d*y + e',strcat('a =
',num2str(a),',b = ',num2str(b),',c = ',num2str(c),',d =
',num2str(c),',e = ',num2str(e)),strcat('focal length = ',
num2str(focal_length),', +- ', num2str(focal_pm))));
        axis equal;
    end
    if(graphics >=1)&(graphics <=2)
        figure(2);
        mesh(x2,y2,z2);
        title(strvcat('z = a*x^2 + b*y^2 + c*x + d*y + e',strcat('a =
',num2str(a),',b = ',num2str(b),',c = ',num2str(c),',d =
',num2str(c),',e = ',num2str(e)),strcat('focal length = ',
num2str(focal_length),', +- ', num2str(focal_pm))));
        hold on;
        plot3(x,y,z,'.');
        axis equal;
        az = 90;
        rotate3d on;
    end
    if(graphics == 2)
        figure(3);
        clf;
        plot3(x,y,z,'b. ');
        hold on;
        plot3([x],[y],[zp],'r. ');
        xlabel('x');
        ylabel('y');
        rotate3d on;
        axis equal;
    end
end

```

Matlab function “ddedemo” used to demonstrate viability of Matlab/PhotoModeler DDE system

```

function ddedemo
close all;
clear all;
%PhotoModeler's DDE commands take a very simple form.
%Only the "DDE Initiate", "DDE Request" and "DDE Terminate" messages
are used.
%All PhotoModeler DDE commands are ASCII text strings and are of the
format:
%"commandName commandParam1 commandParam2 commandParam3 ...".
%All commands return values and a return code indicating success or
failure.
%PhotoModeler's DDE commands are NOT case sensitive. For example,
%the commands "GetNextPoint", "getnextpoint" and "GETNEXTPOINT" are all
equivalent.
%
%*"MP PhotoNum 3DPointID x y"where PhotoNum is a valid photograph number
%(as returned by GetPhotoList), 3DPointID is a valid 3D Point ID

```

```

%(starting at value returned by GetNextPointID), x is the X photo
coordinate
%(upper left is 0,0) in pixels, and y is the Y photo coordinate.
format compact;
%Get the XY data
XYs = opendatatr('stage.txt');
camlid = XYs(1,1);
XY1 = XYs(find(XYs(:,1) == camlid),:);
XY2 = XYs(find(XYs(:,1) ~= camlid),:);
XY1 = ordertr(XY1, 2);
XY2 = ordertr(XY2, 2);
X1 = XY1(:, 3); Y1 = XY1(:,4);
X2 = XY2(:, 3); Y2 = XY2(:,4);
close all;
FIG1 = figure(1);
hold on;
set(FIG1, 'Position', [6 144 790 146]);
subplot(1,3,1)
plot(X1,-Y1, 'o'); axis equal; axis off;
subplot(1,3,2)
plot(X2,-Y2, 'ro'); axis equal; axis off;
%Only the "DDE Initiate", "DDE Request" and "DDE Terminate" messages
are used by PM
%OpenProject fullPathOfPmrFile
disp('NOTE: PhotoModeler must be running');
disp('Press any key to mark points');
figure(1);
pause
disp('Mark points in PhotoModler...');
channel = ddeinit('PhotoModeler','Data');
command0= 'OpenProject d:\Matlab\dde\blank.pmr';
OpenProj = ddereq(channel,command0, [1 1]);
photos = 'blank';
RL = 'ResetLists'; ResetList = ddereq(channel,RL, [1 1]);
GPL = 'GetPhotoList'; photos = ddereq(channel,GPL, [1 1]);
GNP = 'GetNum3DPoints'; NumberOf3DPoints = ddereq(channel,GNP, [1 1]);
GNPID = 'GetNextPointID'; NextPointID = ddereq(channel,GNPID, [1 1]);

%Mark Points
index = 1;
maxindex = (length(X1));
while (index <= maxindex)
    c = strcat('MP 1',{' '}, num2str(index), {' '},num2str(X1(index),7),
{' '},num2str((Y1(index)),7));
    c = char(c); %convert c from an array to a string
    out = ddereq(channel, c);
    c = strcat('MP 2',{' '}, num2str(index), {' '},num2str(X2(index),7),
{' '},num2str((Y2(index)),7));
    c = char(c); %convert c from an array to a string
    out = ddereq(channel, c);
    index = index + 1;
end

disp('Processing the data in PhotoModler...')
ProcessData = ddereq(channel,'Process 13', [1 1])
%Collect 3D data
ResetList = ddereq(channel,RL, [1 1]);

```

```

GNP3D = 'GetNextPoint';
index = 1;
while (index <= maxindex)
    point3D = ddereq(channel,GNP3D, [1 1]);
    point3Dn = str2num(point3D);
    XYZ(index,1) = point3Dn(2); %ID
    XYZ(index,2) = point3Dn(5); %x
    XYZ(index,3) = point3Dn(6); %y
    XYZ(index,4) = point3Dn(7); %z
    if(point3Dn(3) ~= 1)
        %warning(strcat('Point:', num2str(point3Dn(2)), ', is not
processed'));
    end

    index = index + 1;
end
subplot(1,3,3)
plot3(XYZ(:,2),XYZ(:,3),XYZ(:,4), 'k. ');
axis equal; axis off;
rotate3d on; view([122 56]);

if(ddeterm(channel))
    disp('dde session successfully ended');
else
    disp('dde session unsuccessfully ended');
end
subplot(1,3,1)
title('Image points 1')
subplot(1,3,2)
title('Image points 2')
subplot(1,3,3)
title('3D points')
format loose;

```

Matlab function “RealSpace” used to simulate the camera views of a dynamic, 3d object

```

function RealSpace
%function RealSpace, Written by Louis Giersch
%used to simulate the views of two cameras on a dynamic, 3d object

close all;
X = [0 -1.5 1 0 0 1 -1.5 0]; %
Y = [-.2 -.2 -.2 -.2 .2 .2 .2 .2]; % the 3d object
Z = [1 1.5 3 4 4 3 1.5 1]; %
c1 = 4; c2 = 4; % focal lengths of cameras 1,2
M1 = [0 0 -1; 0 1 0; 1 0 0]*[1 0 0; 0 0 1; 0 -1 0]; % rotation matrix
for camera 1
M2 = [1 0 0; 0 0 1; 0 -1 0]*[-1 0 0; 0 1 0; 0 0 -1]; % rotation matrix
for camera 2
X1 = 5; Y1 = 0; Z1 = 2.5; %position of camera 1
X2 = 0; Y2 = 5; Z2 = 2.5; %position of camera 2
a = .1; %initial angle of 3d object
F1 = figure(1);
set(F1, 'Renderer', 'zbuffer');
hold on;
subplot(2,3,1)

```

```

while(1 == 1)
    Rot_mot = [cos(a), sin(a), 0; -sin(a) cos(a) 0; 0 0 1]; % rotate
    XYZ = [X; Y; Z];
    the 3d object
    XYZn = Rot_mot*XYZ;
    X = XYZn(1,:);
    Y = XYZn(2,:);
    Z = XYZn(3,:);
    clf;
    subplot(2,3,1)
    hold on; axis equal; title('input xyz')
    plot3(X, Y, Z);
    plot3(X, Y, Z, '.');
    plot3(X1, Y1, Z1, 'rx'); plot3([0 X1], [0 Y1], [Z1 Z1], 'r');
    plot3(X2, Y2, Z2, 'ro'); plot3([0 X2], [0 Y2], [Z2 Z2], 'r');
    view([1,1,.5])
    axis([-5 2 -5 2 -2 5])
    subplot(2,3,2)
    hold on; title('Camera 1 (x)')
    x1 = -c1*(M1(1,1)*(X-X1) + M1(1,2)*(Y-Y1) + M1(1,3)*(Z-
Z1))./(M1(3,1)*(X-X1) +M1(3,2)*(Y-Y1) + M1(3,3)*(Z-Z1)); %calculate
camera 1 view
    y1 = -c1*(M1(2,1)*(X-X1) + M1(2,2)*(Y-Y1) + M1(2,3)*(Z-
Z1))./(M1(3,1)*(X-X1) +M1(3,2)*(Y-Y1) + M1(3,3)*(Z-Z1)); %
    plot(x1, y1); plot(x1, y1, '.'); axis([-2 2 -2 2]);
    subplot(2,3,3)
    hold on; title('cam 1 check')
    plot(Y,Z); plot(Y,Z, '.');axis([-2 2 0 4]);
    subplot(2,3,5); hold on; title('Camera 2 (o)')
    x2 = -c2*(M2(1,1)*(X-X2) + M2(1,2)*(Y-Y2) + M2(1,3)*(Z-
Z2))./(M2(3,1)*(X-X2) +M2(3,2)*(Y-Y2) + M2(3,3)*(Z-Z2)); %calculate
camera 2 view
    y2 = -c2*(M2(2,1)*(X-X2) + M2(2,2)*(Y-Y2) + M2(2,3)*(Z-
Z2))./(M2(3,1)*(X-X2) +M2(3,2)*(Y-Y2) + M2(3,3)*(Z-Z2)); %
    plot(x2, y2); plot(x2, y2, '.');
    axis([-2 2 -2 2]);
    subplot(2,3,6)
    hold on; title('cam 2 check')
    plot(X,Z); plot(X,Z, '.');axis([-2 2 0 4]);
    pause(.01);
    tic;
    while (toc < .1)
        drawnow;
    end
end
end

```

Matlab function “k_cam1” used to simulate the K1 distortion of a DC290 camera

function k_cam1

*function k_cam1, Written by Louis Giersch

%Used to simulate the K1 distortion of a DC290 camera

```

K1 = 0.00174
K2 = -4.27E-5
P1 = 0.41E-5
P2 = -9.38E-5

```

```

x = 7.537; % In mm
y = 5.04; % In mm
scale = 170/5.04 %inches in object space/mm on CCD
X0 = zeros(10,10);
Y0 = zeros(10,10);
m = 1:1:length(X0(1,:))^2;
n = 1;
while(n <= length(X0(:,1)))
    X0(:,n) = .1*x*(n- .5 - .5*length(X0(:,1)))*ones(length(X0(:,1)),1)
    n = n + 1;
end
X0
pause
n = 1;
while(n <= length(Y0(:,1)))
    Y0(n,:) = .1*y*(n- .5 - .5*length(X0(:,1)))*ones(length(X0(:,1)),1)
    n = n + 1;
end
Y0
close all;
R2 = X0.^2 + Y0.^2;
XK1 = X0 + K1.*X0.*R2;
YK1 = Y0 + K1.*Y0.*R2;
hold on;
plot(X0, Y0, 'k. ');
plot(XK1, YK1, 'ko');
del = ((X0 - XK1).^2 + (Y0 - YK1).^2).^5;
maxdist = max(max(del));
title(strvcat(strcat('K1 = ', num2str(K1))));%, strcat('maximum
distortion (mm) = ',num2str(maxdist,3)), strcat('If the scene is 170
in. wide, the maximum distortion is =',
num2str(scale*maxdist,3),'(in.)') );
axis image;
axis off;

```

Matlab function “LGVS3,” A prototype target tracking videogrammetry system using Matlab/PhotoModeler DDE

```

function LGVS3
%function LGVS3, Written by Louis Giersch
%A prototype target tracking videogrammetry system using
%Matlab/PhotoModeler DDE

```

```

close all; clear all;
disp('opening image files:')
disp('a1.jpg')
a1 = imread('a1.jpg', 'jpg');
disp('a2.jpg')
a2 = imread('a2.jpg', 'jpg');
disp('a3.jpg')
a3 = imread('a3.jpg', 'jpg');
disp('a4.jpg')
a4 = imread('a4.jpg', 'jpg');
disp('a5.jpg')
a5 = imread('a5.jpg', 'jpg');
disp('b1.jpg')
b1 = imread('b1.jpg', 'jpg');

```

```

disp('b2.jpg')
b2 = imread('b2.jpg', 'jpg');
disp('b3.jpg')
b3 = imread('b3.jpg', 'jpg');
disp('b4.jpg')
b4 = imread('b4.jpg', 'jpg');
disp('b5.jpg')
b5 = imread('b5.jpg', 'jpg');

F1 = figure(1); image(a1); title('camera A'); axis image;
F2 = figure(2); image(b1); title('camera B'); axis image;
F4 = figure(4);
set(F1, 'Position', [152 31 647 532]);
set(F2, 'Position', [152 31 647 532]);
set(F4, 'Position', [120 515 104 7]);
[MAxY,MAxX, junk] = size(a1);
BOXs = 30;
GO = 1;
FINDEX = 1;
numpt = 1;
~~~~~Mark Camera A points~~~~~
figure(1)
while(GO == 1)
[lx, ly, donow] = ginput(1);
hold on;
if(lx + BOXs + 1) > MAxX
    lx = MAxX - BOXs - 1; end
if(lx - BOXs - 1) < 0
    lx = BOXs + 1; end
if(ly + BOXs + 1) > MAxY
    ly = MAxY - BOXs - 1; end
if(ly - BOXs - 1) < 0
    ly = BOXs + 1; end

    if donow > 1.5
        GO = 0;
    else
        CamAlguess(numpt, :) = [lx, ly]; numpt = numpt + 1;
        plot([lx+BOXs, lx+BOXs, lx-BOXs, lx-BOXs, lx+BOXs], [ly+BOXs, ly-
BOXs, ly-BOXs, ly+BOXs, ly+BOXs], 'g'); end
end

~~~~~Mark Camera B points~~~~~
figure(2); pause(1); donow = 1;
GO = 1; FINDEX = 1; numpt = 1;
while(GO == 1)
[lx, ly, donow] = ginput(1);
hold on;
if(lx + BOXs + 1) > MAxX
    lx = MAxX - BOXs - 1; end
if(lx - BOXs - 1) < 0
    lx = BOXs + 1; end
if(ly + BOXs + 1) > MAxY
    ly = MAxY - BOXs - 1; end
if(ly - BOXs - 1) < 0
    ly = BOXs + 1; end
    if donow == 3

```

```

donow2 = input('type 2 to track:');
    if donow2 == 2
        GO = 0; end
    if donow2 == 3
        GO = 0; break; end
else
    CamB1guess(numpt, :) = [lx, ly]; numpt = numpt + 1;
    plot([lx+BOXs, lx+BOXs, lx-BOXs, lx-BOXs, lx+BOXs], [ly+BOXs, ly-
BOXs, ly-BOXs, ly+BOXs, ly+BOXs], 'g'); end
end

A1points = LGtrack(a1, CamA1guess, numpt, BOXs, 1);
B1points = LGtrack(b1, CamB1guess, numpt, BOXs, 2);
channel = DDEinit; XYZ1 = DDE3d(A1points, B1points, channel);
DDEend(channel);

A2points = LGtrack(a2, A1points, numpt, BOXs, 1);
B2points = LGtrack(b2, B1points, numpt, BOXs, 2);
channel = DDEinit; XYZ2 = DDE3d(A1points, B1points, channel);
DDEend(channel);

A3points = LGtrack(a3, A2points, numpt, BOXs, 1);
B3points = LGtrack(b3, B2points, numpt, BOXs, 2);
channel = DDEinit; XYZ3 = DDE3d(A1points, B1points, channel);
DDEend(channel);

A4points = LGtrack(a4, A3points, numpt, BOXs, 1);
B4points = LGtrack(b4, B3points, numpt, BOXs, 2);
channel = DDEinit; XYZ4 = DDE3d(A1points, B1points, channel);
DDEend(channel);

A5points = LGtrack(a5, A4points, numpt, BOXs, 1);
B5points = LGtrack(b5, B4points, numpt, BOXs, 2);
channel = DDEinit; XYZ5 = DDE3d(A1points, B1points, channel);
DDEend(channel);

F3 = figure(3); clf; title('3D'); axis image;
set(F3, 'Position', [152 31 647 532]); set(F3, 'Renderer', 'zbuffer')
while (1 == 1)
    plot3(XYZ1(:,1), XYZ1(:,2), XYZ1(:,3), 'k. '); view([-74.5000 14]);
    pause(.01); drawnow;
    plot3(XYZ2(:,1), XYZ2(:,2), XYZ2(:,3), 'k. '); view([-74.5000 14]);
    pause(.01); drawnow;
    plot3(XYZ3(:,1), XYZ3(:,2), XYZ3(:,3), 'k. '); view([-74.5000 14]);
    pause(.01); drawnow;
    plot3(XYZ4(:,1), XYZ4(:,2), XYZ4(:,3), 'k. '); view([-74.5000 14]);
    pause(.01); drawnow;
    plot3(XYZ5(:,1), XYZ5(:,2), XYZ5(:,3), 'k. '); view([-74.5000 14]);

    pause(.01); drawnow;
    plot3(XYZ4(:,1), XYZ4(:,2), XYZ4(:,3), 'k. '); view([-74.5000 14]);
    pause(.01); drawnow;
    plot3(XYZ3(:,1), XYZ3(:,2), XYZ3(:,3), 'k. '); view([-74.5000 14]);
    pause(.01); drawnow;
    plot3(XYZ2(:,1), XYZ2(:,2), XYZ2(:,3), 'k. '); view([-74.5000 14]);
end

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function newxy = LGtrack(IM, oldxy, numpt, BOXs, fign)
figure(fign)
image(IM);
JJIND = 1;
FINDEX = 1;
while JJIND < numpt
lx = oldxy(JJIND, 1); ly = oldxy(JJIND, 2);
plot([lx+BOXs, lx+BOXs, lx-BOXs, lx-BOXs, lx+BOXs], [ly+BOXs, ly-BOXs,
ly-BOXs, ly+BOXs, ly+BOXs], 'g');
JJIND = JJIND + 1;
lx = round(lx); ly = round(ly);
ac = lx - BOXs:1:lx + BOXs; ar = ly - BOXs:1:ly + BOXs;
area = double(IM(ar, ac));
[xout, yout] = LGVS_cent(area);
xout = xout + lx - BOXs; yout = yout + ly - BOXs;
figure(fign); plot(xout, yout, 'r+'); zoom out; drawnow;
newxy(FINDEX, :) = [xout yout];
FINDEX = FINDEX + 1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function channel = DDEinit
disp('Initiating DDE session...');
channel = ddeinit('PhotoModeler','Data');
command0 = 'OpenProject d:\Matlab\dde\blank290.pmr';
OpenProj = ddereq(channel,command0, [1 1]);
photos = 'blank';
RL = 'ResetLists'; ResetList = ddereq(channel,RL, [1 1]);
GPL = 'GetPhotoList'; photos = ddereq(channel,GPL, [1 1]);
GNP = 'GetNum3DPoints'; NumberOf3DPoints = ddereq(channel,GNP, [1 1]);
GNPID = 'GetNextPointID'; NextPointID = ddereq(channel,GNPID, [1 1]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function XYZ = DDE3d(Axy, Bxy, channel)
index = 1;
RL = 'ResetLists'; GPL = 'GetPhotoList'; GNP = 'GetNum3DPoints'; GNPID
= 'GetNextPointID';
maxindex = (length(Axy(:,1)));
while (index <= maxindex)
c = strcat('MP 1',{' '}, num2str(index), {'
'},num2str(Axy(index,1),7), {' '},num2str((Axy(index,2)),7));
c = char(c); %convert c from an array to a string
out = ddereq(channel, c);
c = strcat('MP 2',{' '}, num2str(index), {'
'},num2str(Bxy(index,1),7), {' '},num2str((Bxy(index,2)),7));
c = char(c); %convert c from an array to a string
out = ddereq(channel, c);
index = index + 1;
end
disp('Processing the data in PhotoModler...')
ProcessData = ddereq(channel,'Process 13', [1 1])
%Collect 3D data

```

```

ResetList = ddereq(channel,RL, [1 1]);
GNP3D = 'GetNextPoint';
index = 1;
while (index <= maxindex)
    point3D = ddereq(channel,GNP3D, [1 1]);
    point3Dn = str2num(point3D);
    XYZ(index,1) = point3Dn(5); %x
    XYZ(index,2) = point3Dn(6); %y
    XYZ(index,3) = point3Dn(7); %z
    if(point3Dn(3) ~= 1)
        %warning(strcat('Point:', num2str(point3Dn(2)), '. is not
processed'));
    end
    index = index + 1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function DDEend(channel)
if(ddeterm(channel))
    disp('dde session successfully ended');
else
    disp('dde session unsuccessfully ended');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE November 2001		3. REPORT TYPE AND DATES COVERED Contractor Report
4. TITLE AND SUBTITLE Pathfinder Photogrammetry Research for Ultra-Lightweight and Inflatable Space Structures			5. FUNDING NUMBERS CA NCC1-01017 WU 755-06-00-03	
6. AUTHOR(S) Louis Roy Miller Giersch				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) The George Washington University Joint Institute for Advancement of Flight Sciences NASA Langley Research Center Hampton, VA 23681-2199			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-2199			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA/CR-2001-211244	
11. SUPPLEMENTARY NOTES The information presented in this report was submitted to the School of Engineering and Applied Science of The George Washington University in partial fulfillment of the requirements for the Degree of Master of Science, August 2001. Langley Technical Monitor: Richard S. Pappa				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 39 Distribution: Standard Availability: NASA CASI (301) 621-0390			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The defining characteristic of ultra-lightweight and inflatable space structures is that they are both very large and very low mass. This makes standard contacting methods of measurement (e.g. attaching accelerometers) impractical because the dynamics of the structure would be changed by the mass of the contacting instrument. Optical measurements are therefore more appropriate. Photogrammetry is a leading candidate for the optical analysis of gossamer structures because it allows for the measurement of a large number of points, is amenable to time sequences, and offers the potential for a high degree of accuracy. The purpose of this thesis is to develop the methodology and determine the effectiveness of a photogrammetry system in measuring ultra-lightweight and inflatable space structures. The results of this thesis will be considered in the design of an automated photogrammetry system for the 16m-diameter vacuum chamber at the NASA Langley Research Center.				
14. SUBJECT TERMS Photogrammetry, Optical 3-D Measurement Techniques, Ultra-Lightweight and Inflatable Structures, Large Space Structures, Gossamer Spacecraft			15. NUMBER OF PAGES 76	
			16. PRICE CODE A05	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	
